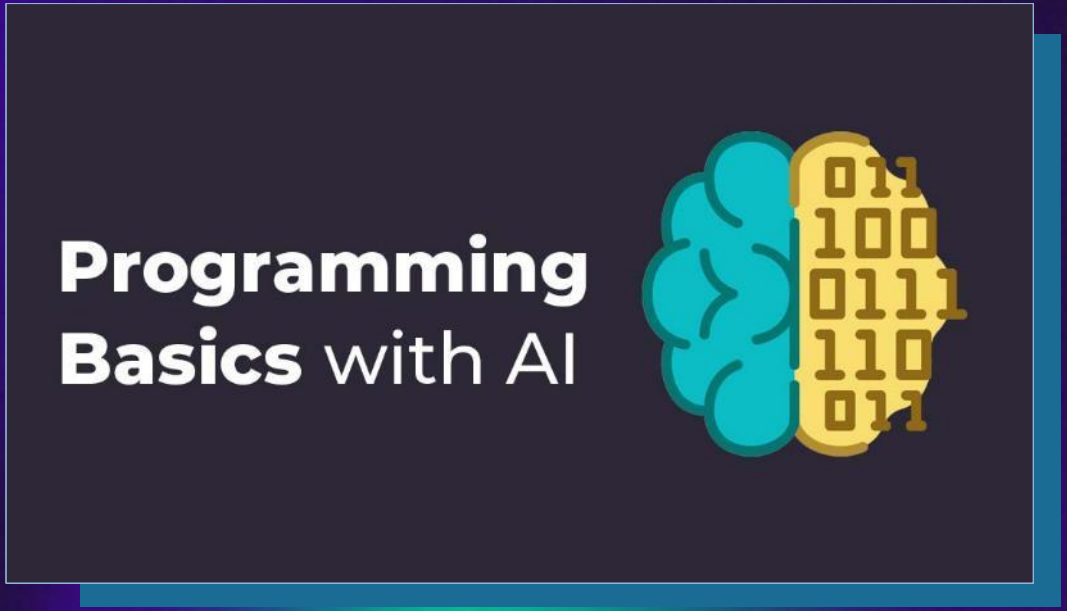
Intro to Coding with AI

Using VS Code, Variables, Data, Calculations, Functions, Quick Intro to HTML, CSS and JS



Svetlin Nakov, PhD

Co-founder @ SoftUni

SoftUni AI

https://ai.softuni.bg

Agenda

SoftUni AI

1. Intro to **coding**: **writing** and **running** code
2. **Data**, **calculations** and **functions** in JavaScript
3. **SoftUni Judge**: automated code evaluation system
4. Working with **text** and **numbers** in JS

Sli.do Code

#AI-Programming

Join at

slido.com

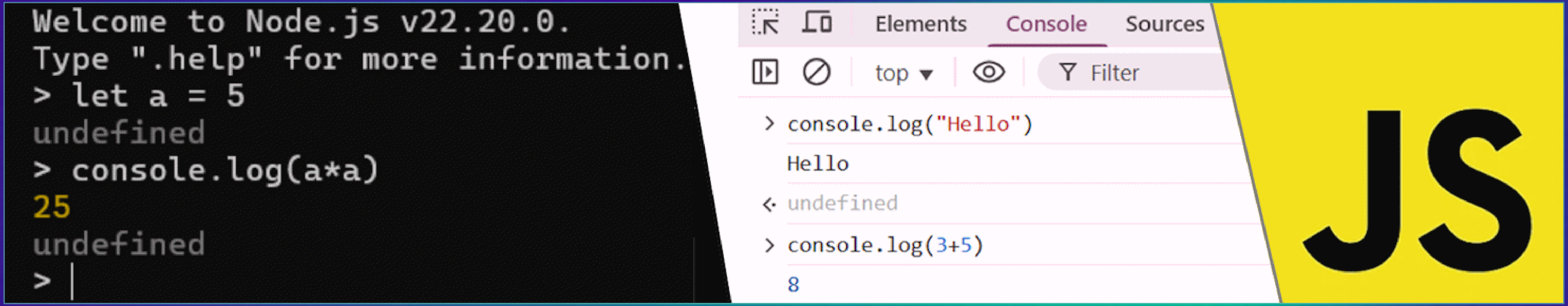
#AI-Programming

Breaks

20:00 / 21:00

Intro to Coding

Coding, JavaScript, Running JS Code in the Web Browser, Node.js and VS Code



```
Welcome to Node.js v22.20.0.
Type ".help" for more information.
> let a = 5
undefined
> console.log(a*a)
25
undefined
> |
```

```
> console.log("Hello")
  Hello
<· undefined
> console.log(3+5)
  8
```

Welcome to Coding

- **Coding** == giving commands to the computer
  - Using a **programming language**, like JavaScript
- Sample **commands**:

`console.log("Hello")` → `Hello`

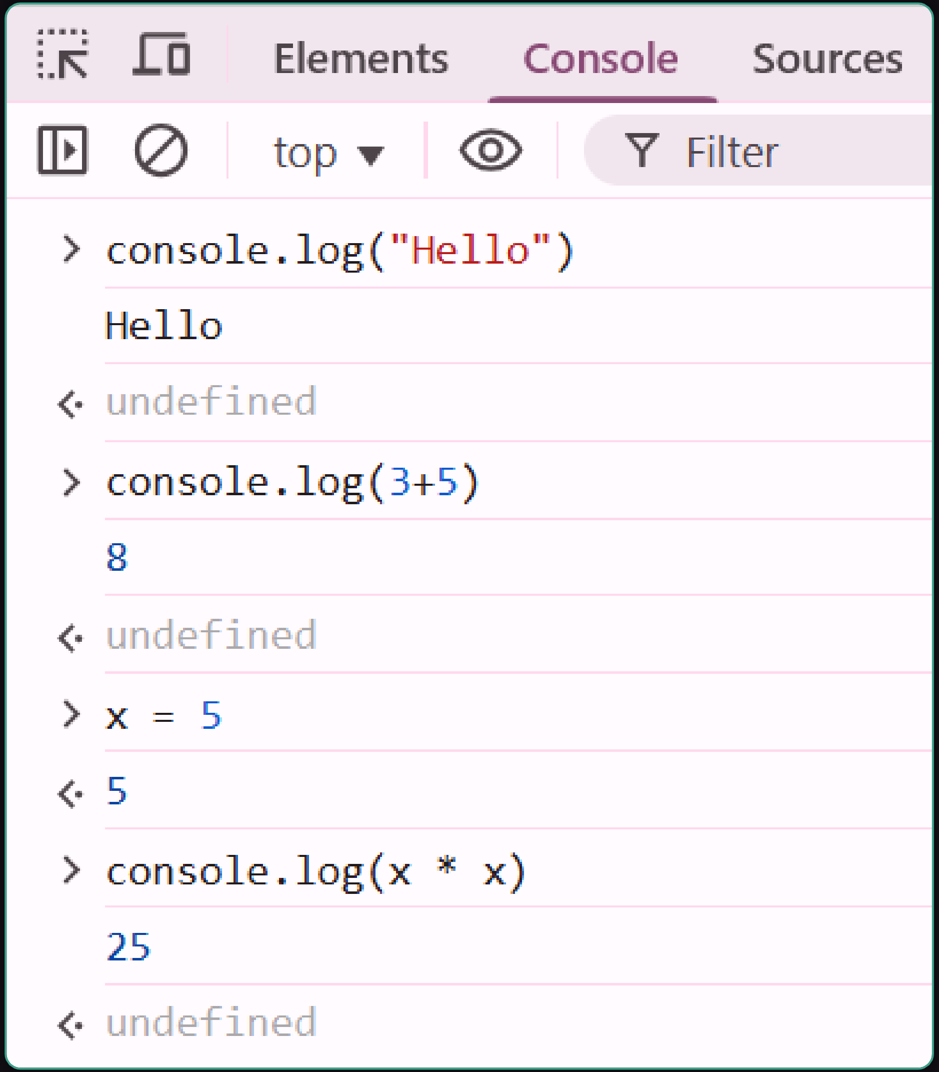
`console.log(3+5)` → `8`

`x = 5` → `5`

`console.log(x * x)` → `25`

Running Code in the Browser

- Web browsers have a built-in **JavaScript** runtime
- Open the **Developer Console**: press **[F12]** key (in Windows)
  - **[F12]** works on **Chrome** / **Firefox**
  - On Mac, use **[Cmd ⌘ + Alt ⌥ + I]**
  - **Safari**: View → Developer → Developer Tools
- Select the **[Console]** tab and type your commands



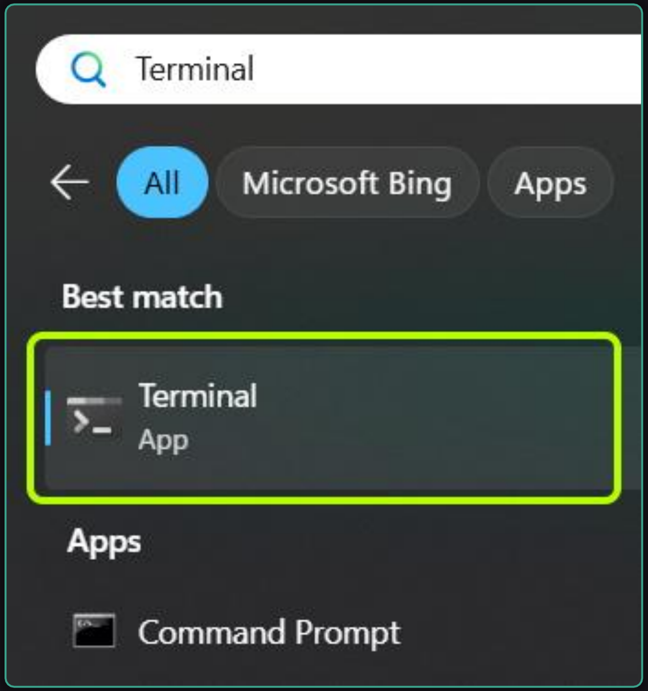
```
> console.log("Hello")
  Hello
<· undefined
> console.log(3+5)
  8
<· undefined
> x = 5
<· 5
> console.log(x * x)
  25
<· undefined
```

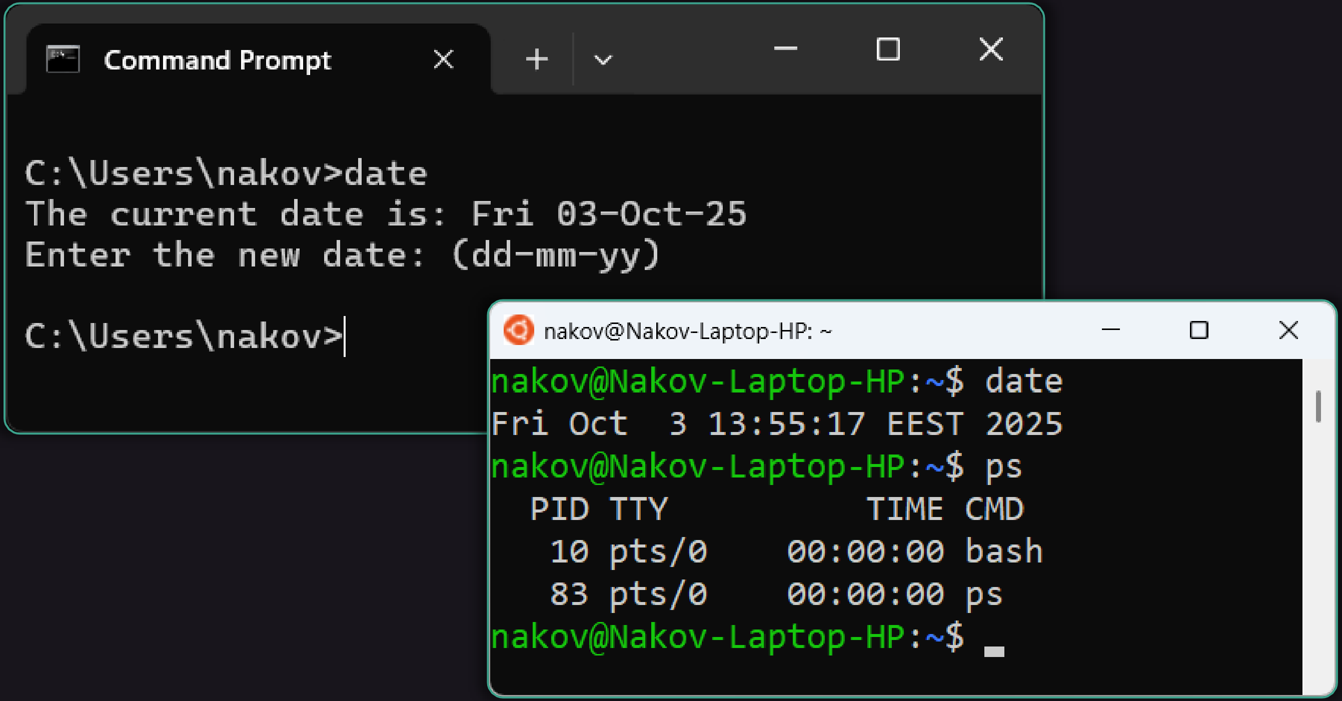
Running Commands in the Web Browser

Live Demo

Welcome to the System Terminal

- The **terminal** is a **command-line interface** (CLI) to invoke system commands (e. g. list files, start an app, …)
- Open the system **Terminal** (in Windows / Linux / Mac)



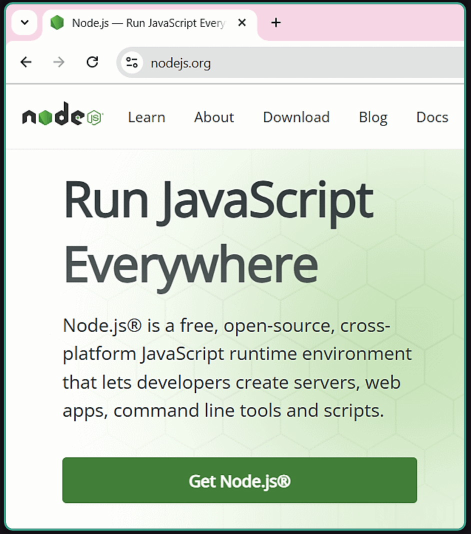


```
C:\Users\nakov>date
The current date is: Fri 03-Oct-25
Enter the new date: (dd-mm-yy)

C:\Users\nakov>
```

```
nakov@Nakov-Laptop-HP:~$ date
Fri Oct  3 13:55:17 EEST 2025
nakov@Nakov-Laptop-HP:~$ ps
  PID TTY          TIME CMD
   10 pts/0    00:00:00 bash
   83 pts/0    00:00:00 ps
nakov@Nakov-Laptop-HP:~$ 
```

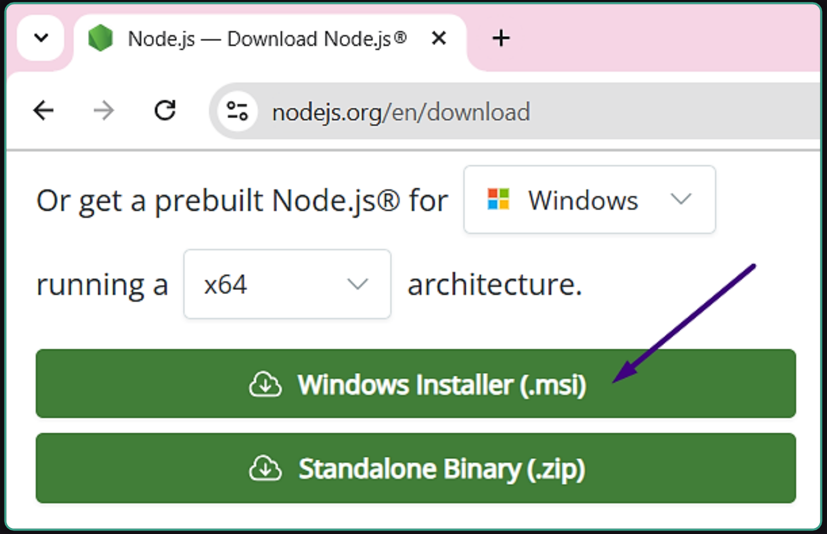
Welcome to Node.js

- **Node.js** is a local **JavaScript runtime** for developers
  - It **runs JS code** and **apps** on your laptop
  - Download & install from: https://nodejs.org

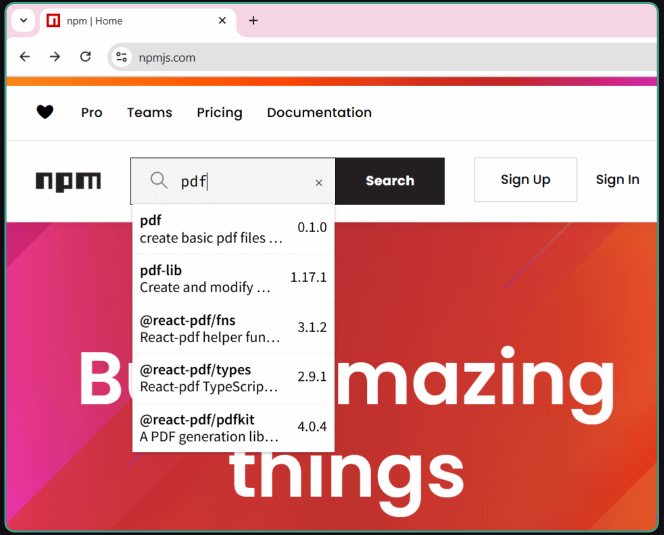


Run JavaScript Everywhere

Node.js® is a free, open-source, cross-platform JavaScript runtime environment that lets developers create servers, web apps, command line tools and scripts.



Or get a prebuilt Node.js® for Windows running a x64 architecture.



pdf

pdf create basic pdf files …	0.1.0
pdf-lib Create and modify …	1.17.1
@react-pdf/fns React-pdf helper fun…	3.1.2
@react-pdf/types React-pdf TypeScrip…	2.9.1
@react-pdf/pdfkit A PDF generation lib…	4.0.4

  - [NPM](NPM) – the biggest repository of **JS libraries** and dev tools

Running JS Code in the Terminal

- Start the **Node.js interpreter** in the Terminal:

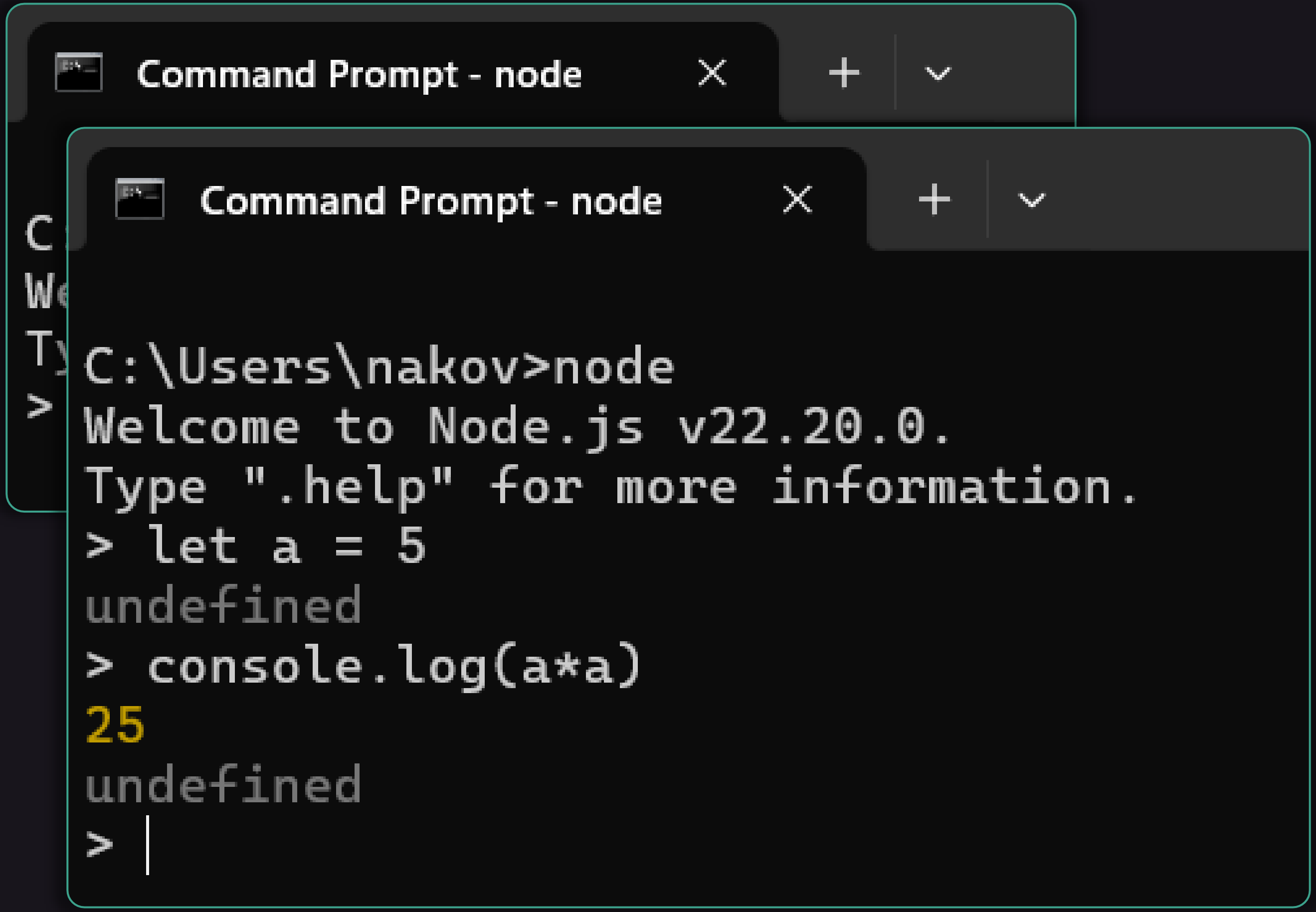
```
node
```

- Run a few JavaScript commands:

```
let a = 5;
```

```
console.log(a*a);
```

```
console.log(a**10);
```



```
C:\Users\nakov>node
Welcome to Node.js v22.20.0.
Type ".help" for more information.
> let a = 5
undefined
> console.log(a*a)
25
undefined
> 
```

Running Commands in Node.js

Live Demo

Welcome to JavaScript

SoftUni AI

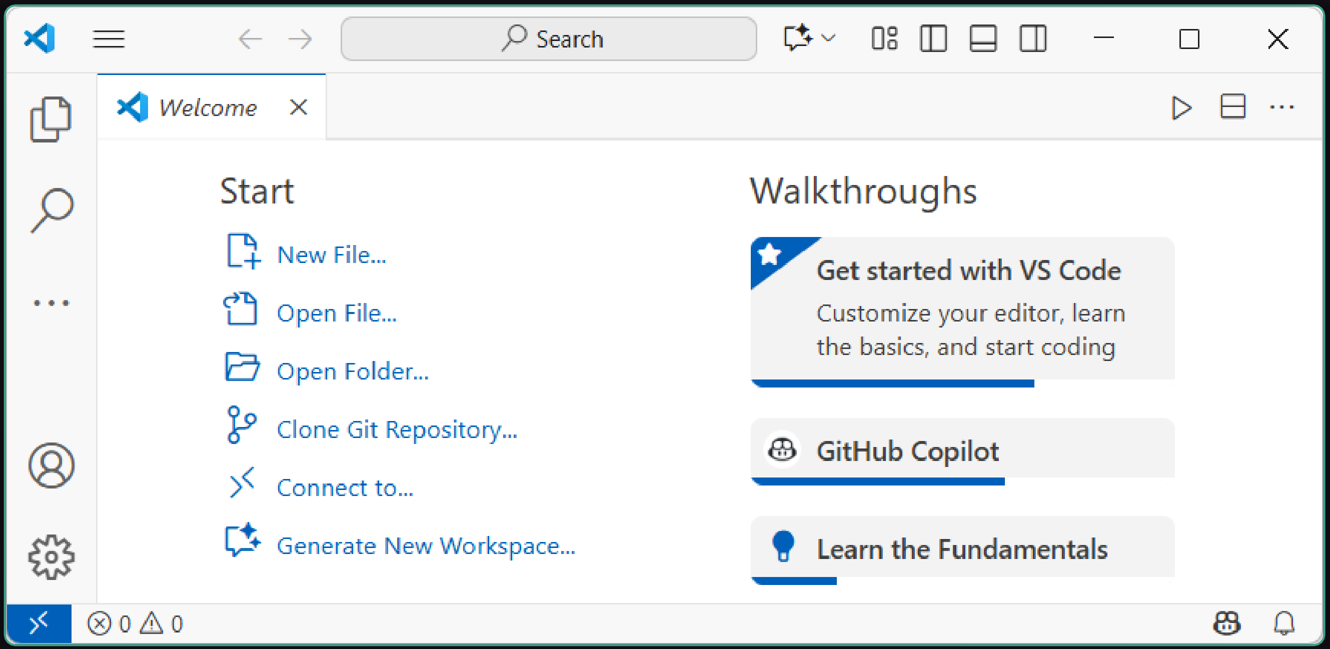
- JavaScript (JS) is a modern **programming language**
  - Extremely **popular**, very large community
  - **Easy to learn**, read, write and understand
  - Lightweight scripting language
  - Executed in a **JavaScript runtime** (JS interpreter) like Web browser or Node.js
- JS is the best language for **applied programming**
- JS is the **recommended** choice for **vibe coders**

Why We Teach JavaScript?

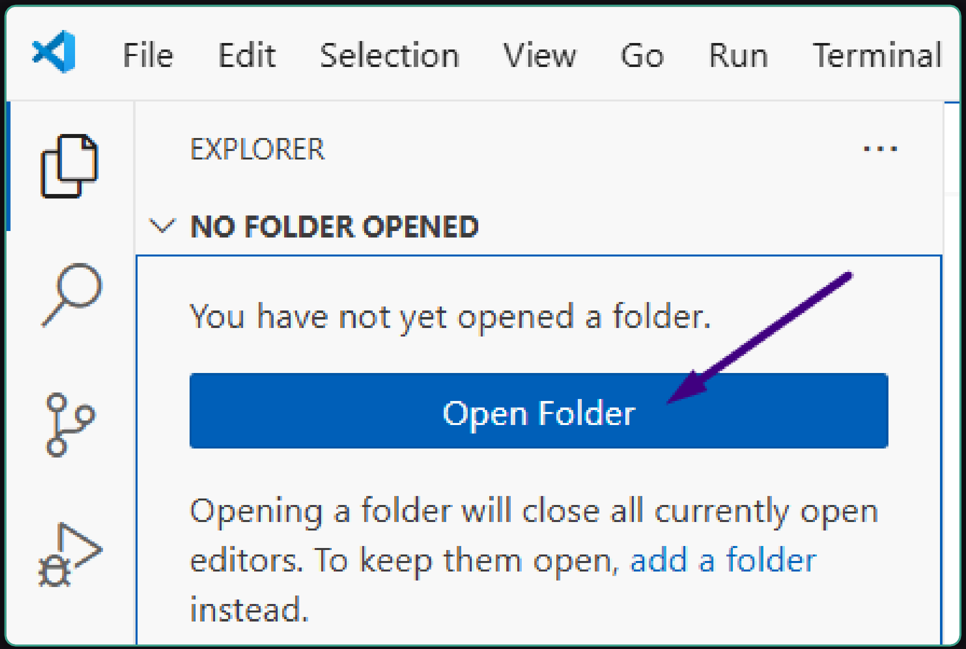
- Why we **start with JavaScript** at SoftUni?
- JavaScript is **universal**: build Web apps, front-end, back-end, mobile apps, games, AI apps, embedded apps, …
- **JS** and **Python** are the most popular languages (in 2025)
- **Python** is **back-end only** (server-side language)!
  - **Python**, **Java**, **C#**, **PHP**, **C++** are **server-side languages** → you still need to code in JS for the front-end (user interface)
- **Python** is best for science, **AI** and **machine learning**

Creating a Project in VS Code

- Start **VS Code**
- Open a **new window**: press **[Shift + Ctrl + N]** keys
- Create a **new folder "Intro-Coding"** and **open** the folder



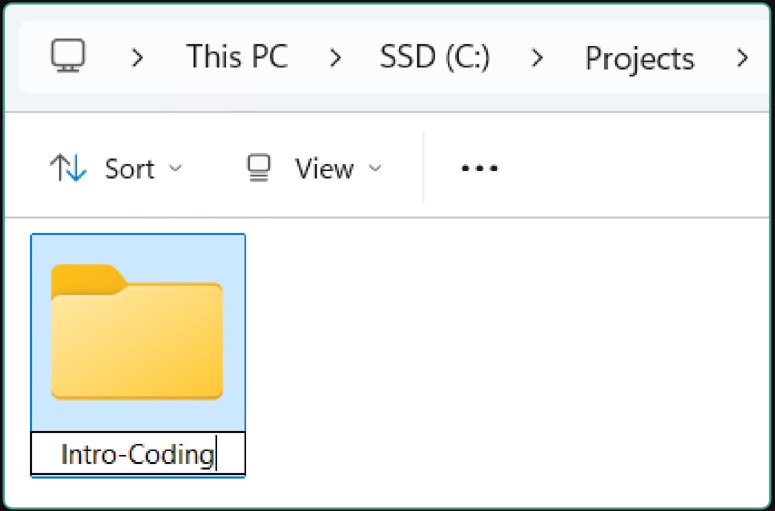
Customize your editor, learn the basics, and start coding



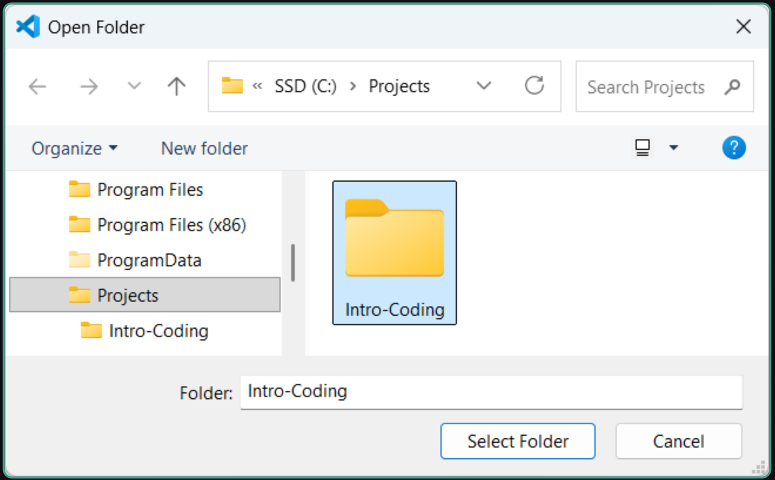
You have not yet opened a folder.

Opening a folder will close all currently open editors. To keep them open, add a folder instead.

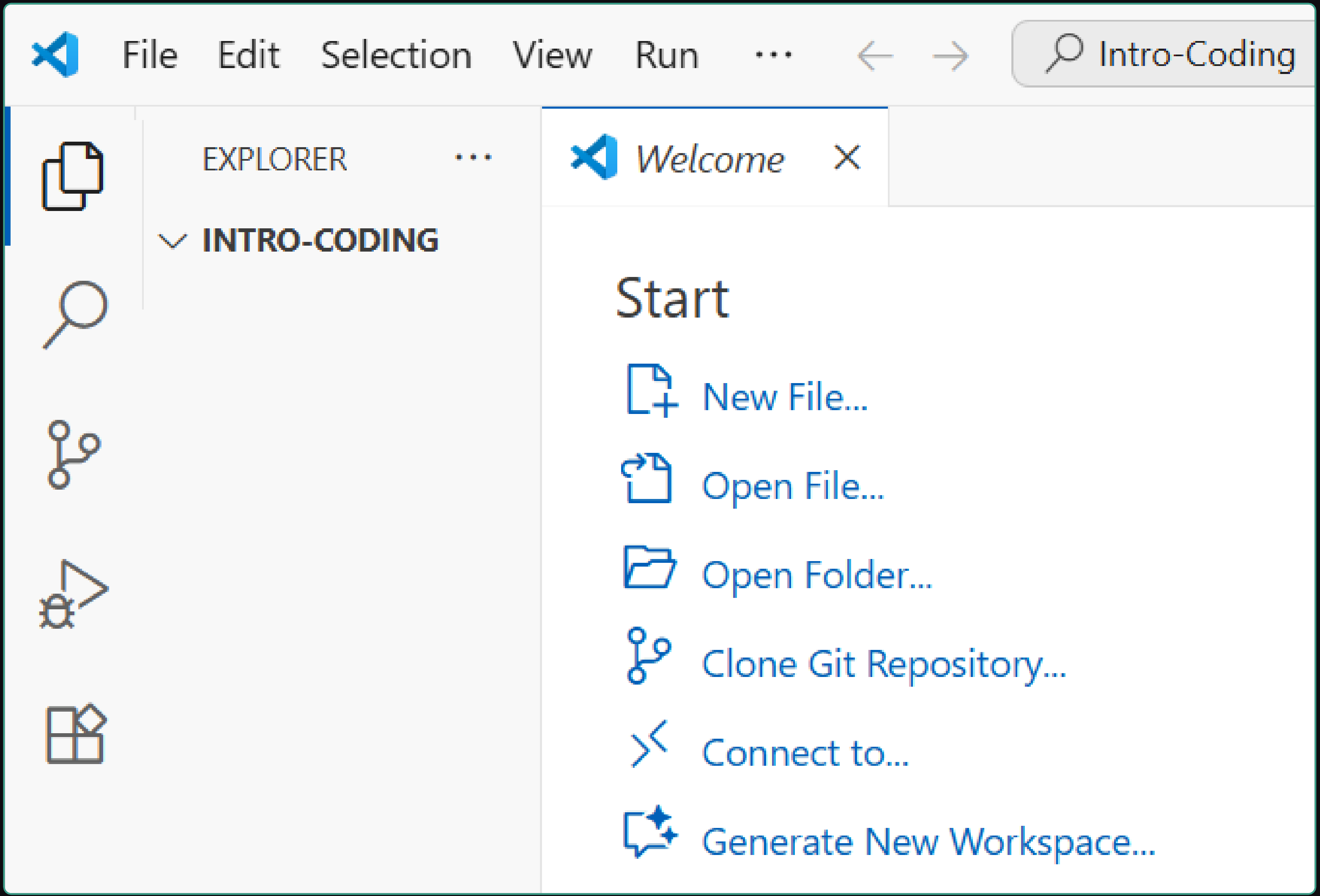
Creating a Project in VS Code (2)



Intro-Coding



Folder: Intro-Coding

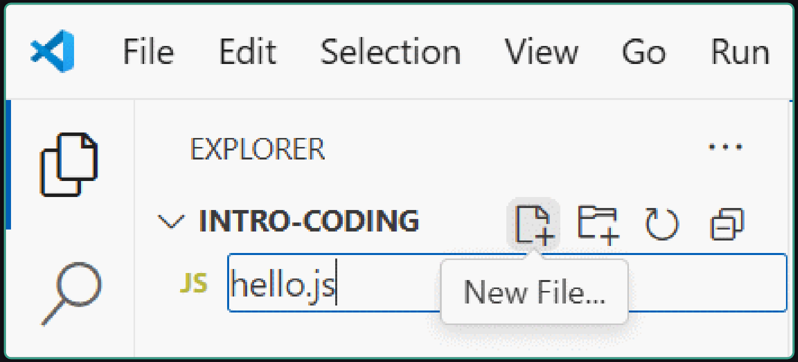


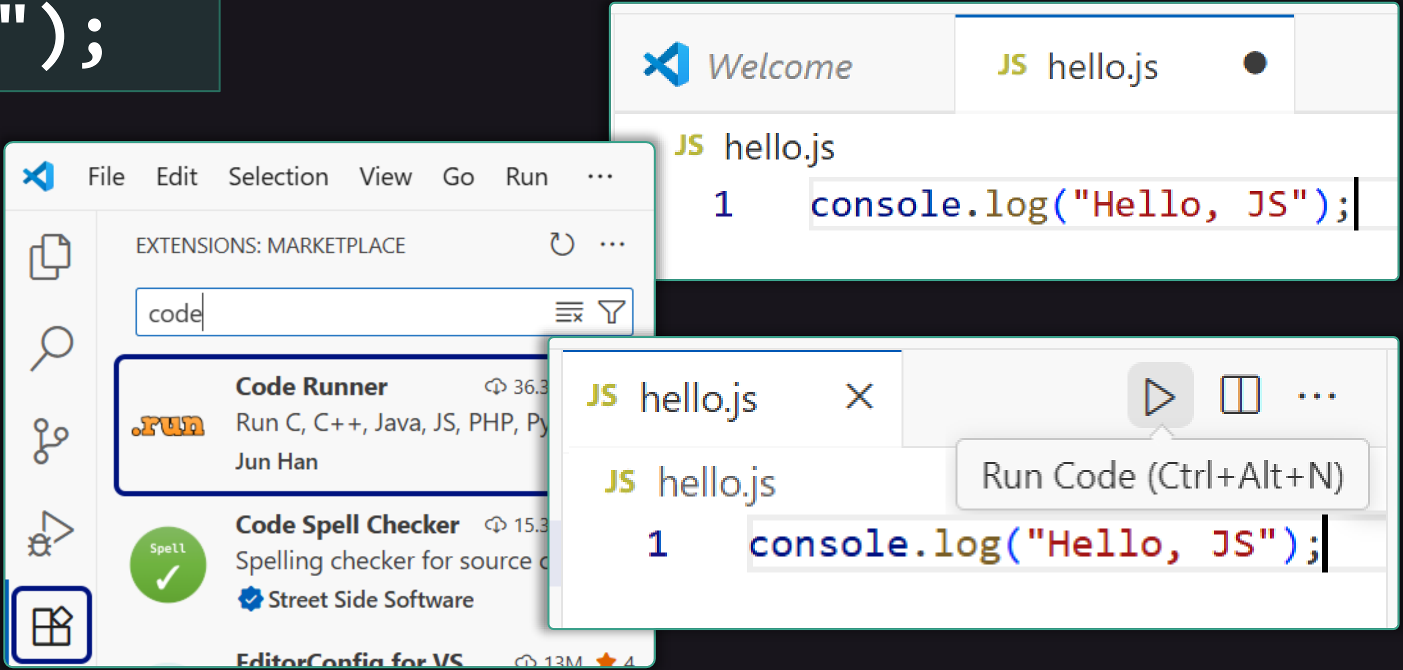
Running JS Code in VS Code

- Create a new JavaScript file inside your project: **hello.js**
- Type a sample JS command:

```
console.log("Hello JS");
```

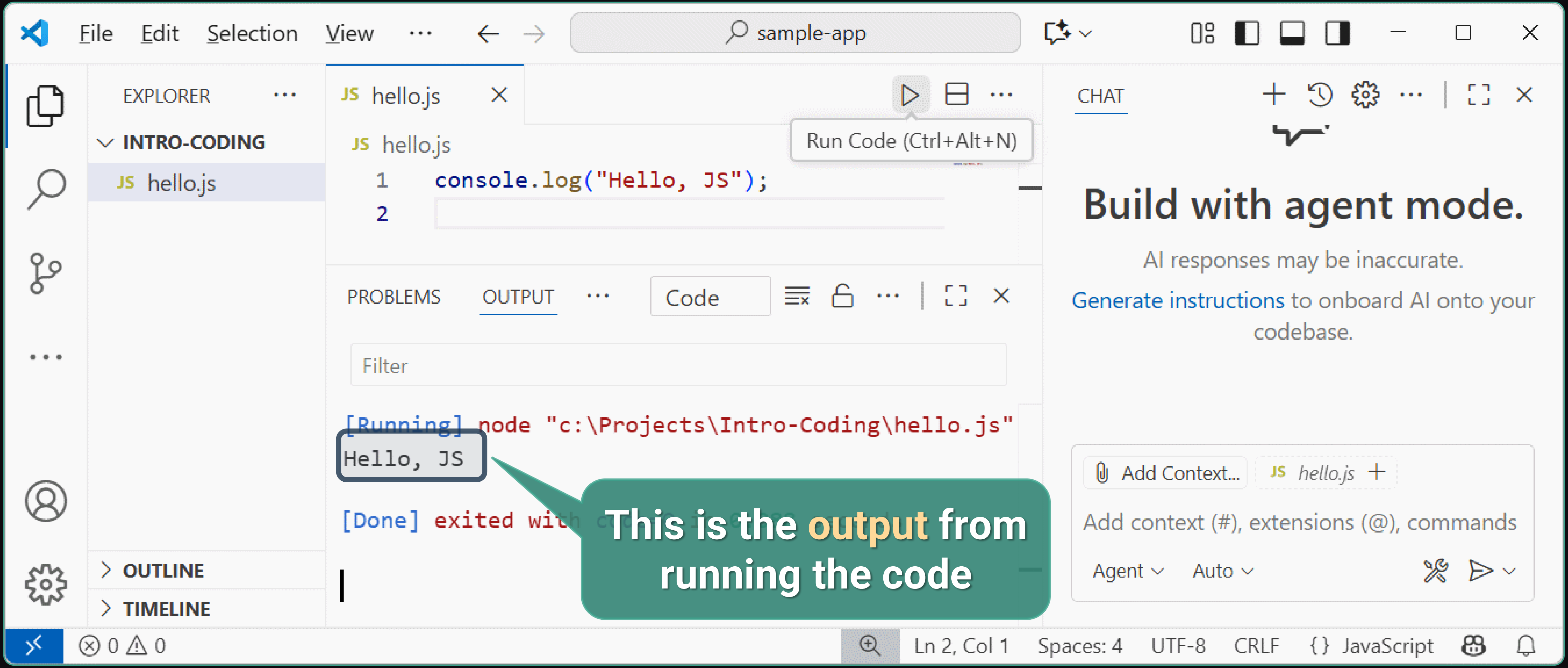
- Install "**Code Runner**" extension in VS Code
- Press **[Run Code]** button





```
console.log("Hello, JS");
```

Running JS Code in VS Code (2)



```javascript
console.log("Hello, JS");
```

```
[Running] node "c:\Projects\Intro-Coding\hello.js"
Hello, JS

[Done] exited with co
```

Build with agent mode.

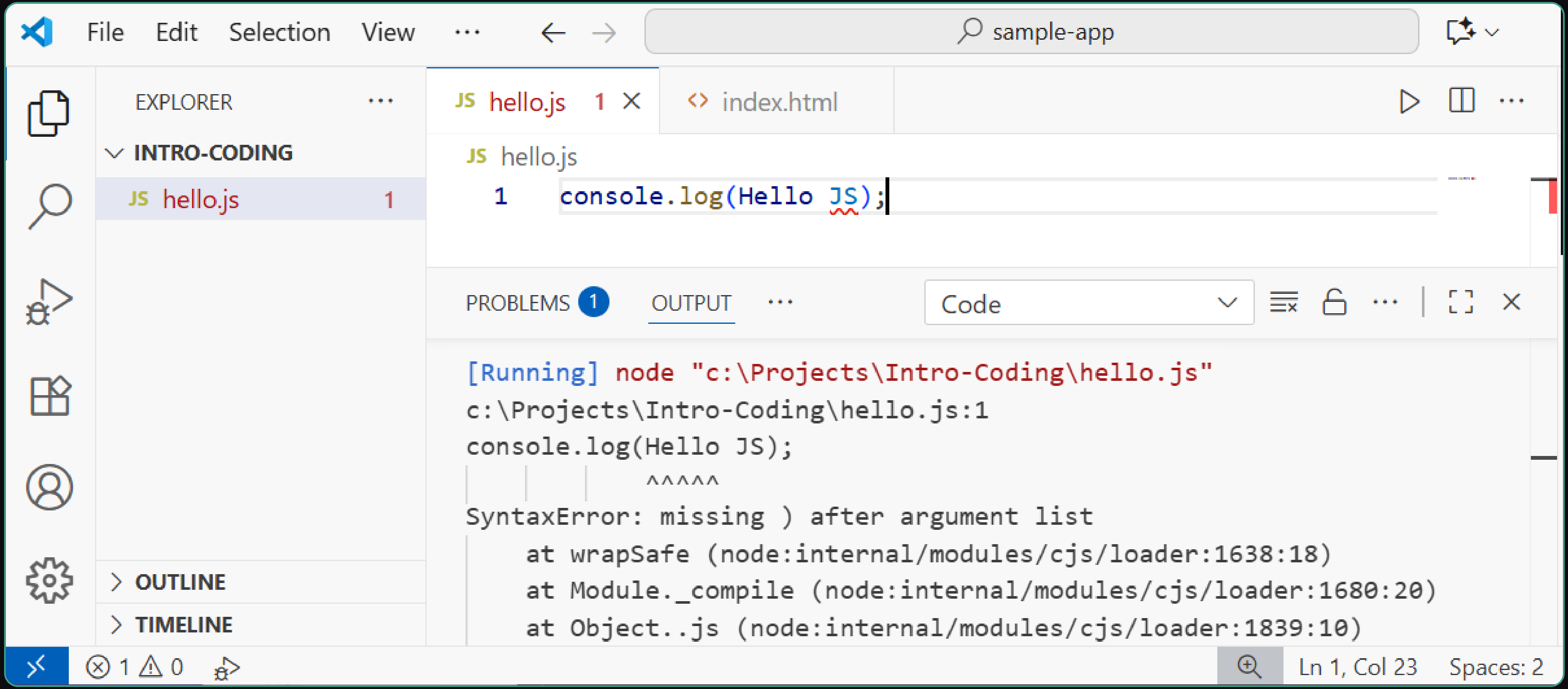
AI responses may be inaccurate.

Generate instructions to onboard AI onto your codebase.

This is the **output** from running the code

Errors in the Code

- Code may have **errors** (syntax error, logical bug)
- Running code with errors produces an **error message**

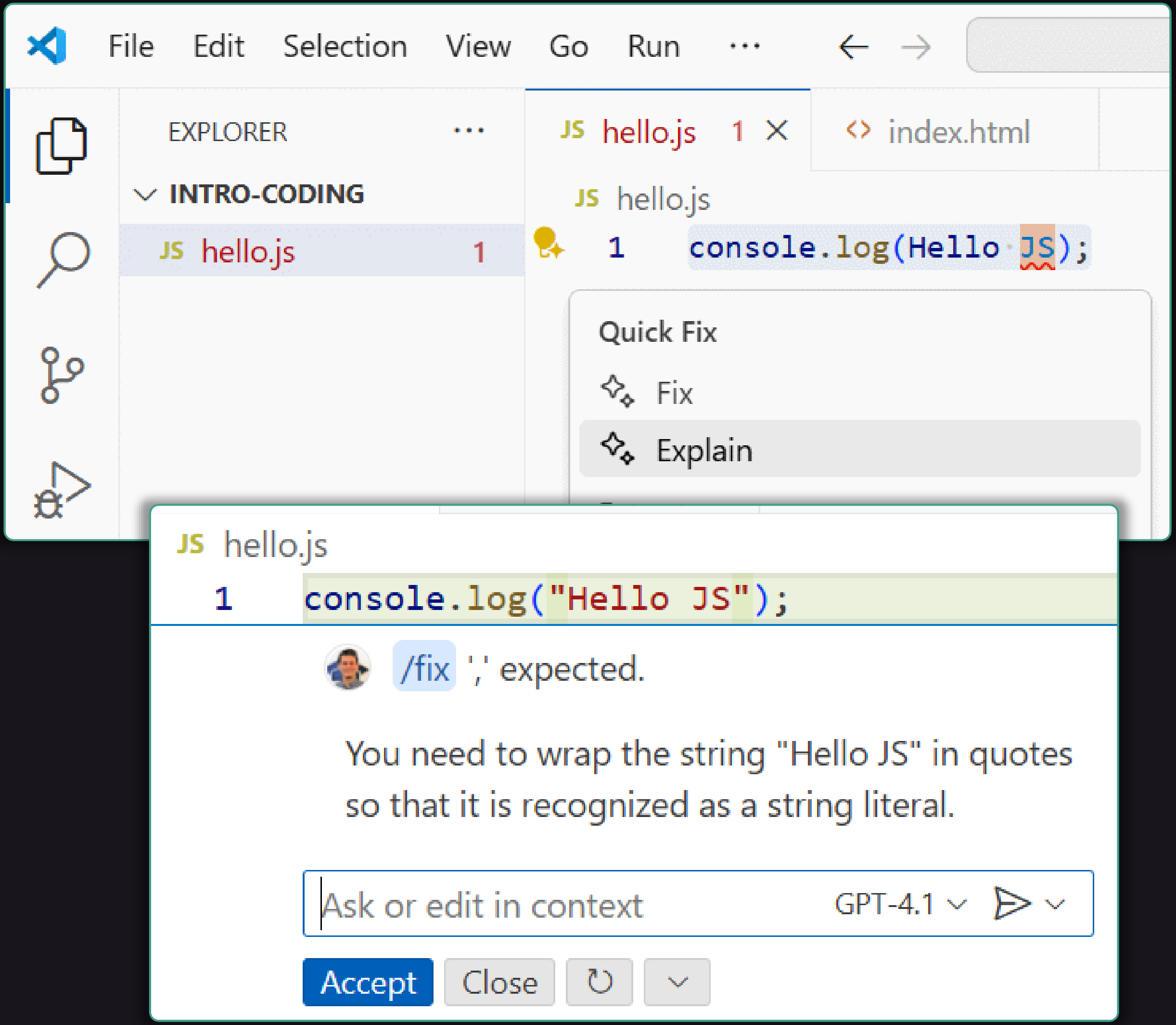


```
console.log(Hello JS);
```

```
[Running] node "c:\Projects\Intro-Coding\hello.js"
c:\Projects\Intro-Coding\hello.js:1
console.log(Hello JS);
            ^^^^^
SyntaxError: missing ) after argument list
    at wrapSafe (node:internal/modules/cjs/loader:1638:18)
    at Module._compile (node:internal/modules/cjs/loader:1680:20)
    at Object..js (node:internal/modules/cjs/loader:1839:10)
```

Explain / Fix an Error

SoftUni AI

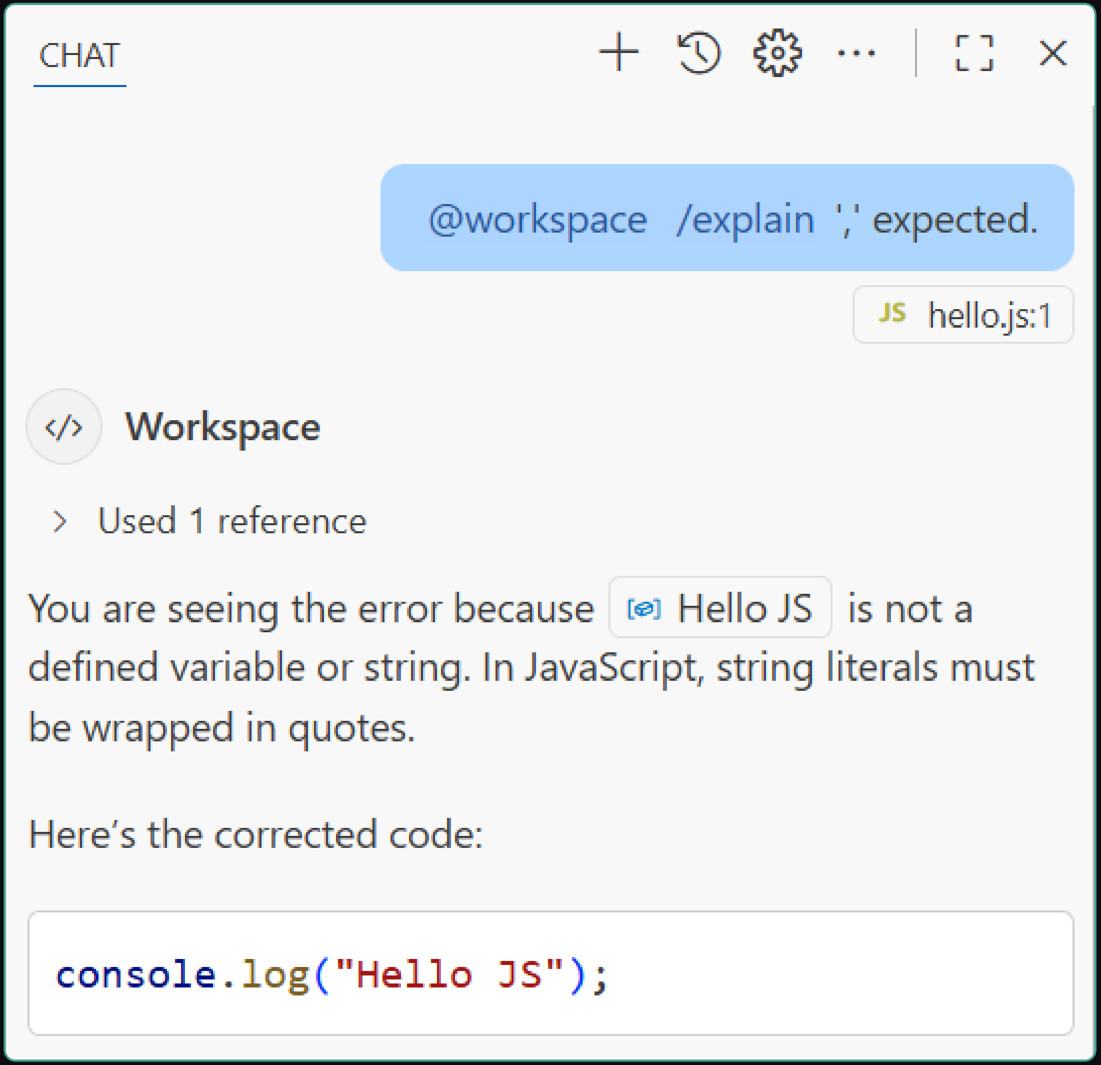


```
console.log(Hello JS);
```

```
console.log("Hello JS");
```

/fix ',' expected.

You need to wrap the string "Hello JS" in quotes so that it is recognized as a string literal.



@workspace /explain ',' expected.

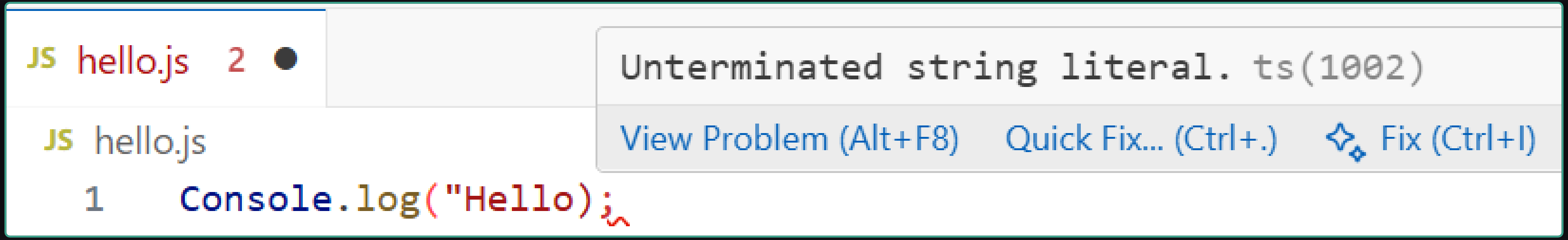
You are seeing the error because Hello JS is not a defined variable or string. In JavaScript, string literals must be wrapped in quotes.

Here’s the corrected code:

```
console.log("Hello JS");
```

Typical Syntax Errors / Bugs

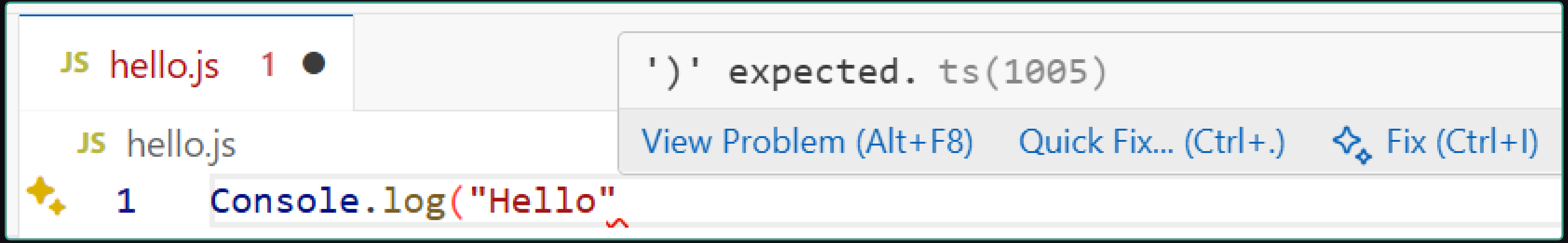
- Unclosed quote → syntax error



```
Unterminated string literal. ts(1002)
```

```
Console.log("Hello);
```

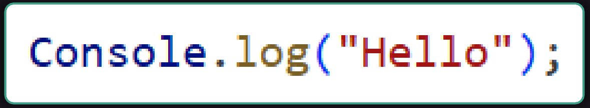
- Unclosed bracket → syntax error



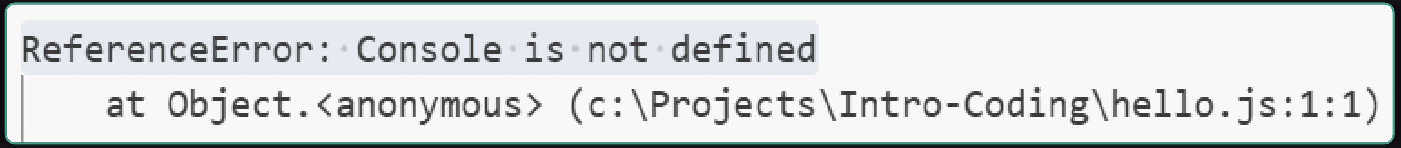
```
')' expected. ts(1005)
```

```
Console.log("Hello"
```

- Capital / small letter interchange → runtime error



```
Console.log("Hello");
```



```
ReferenceError: Console is not defined
    at Object.<anonymous> (c:\Projects\Intro-Coding\hello.js:1:1)
```

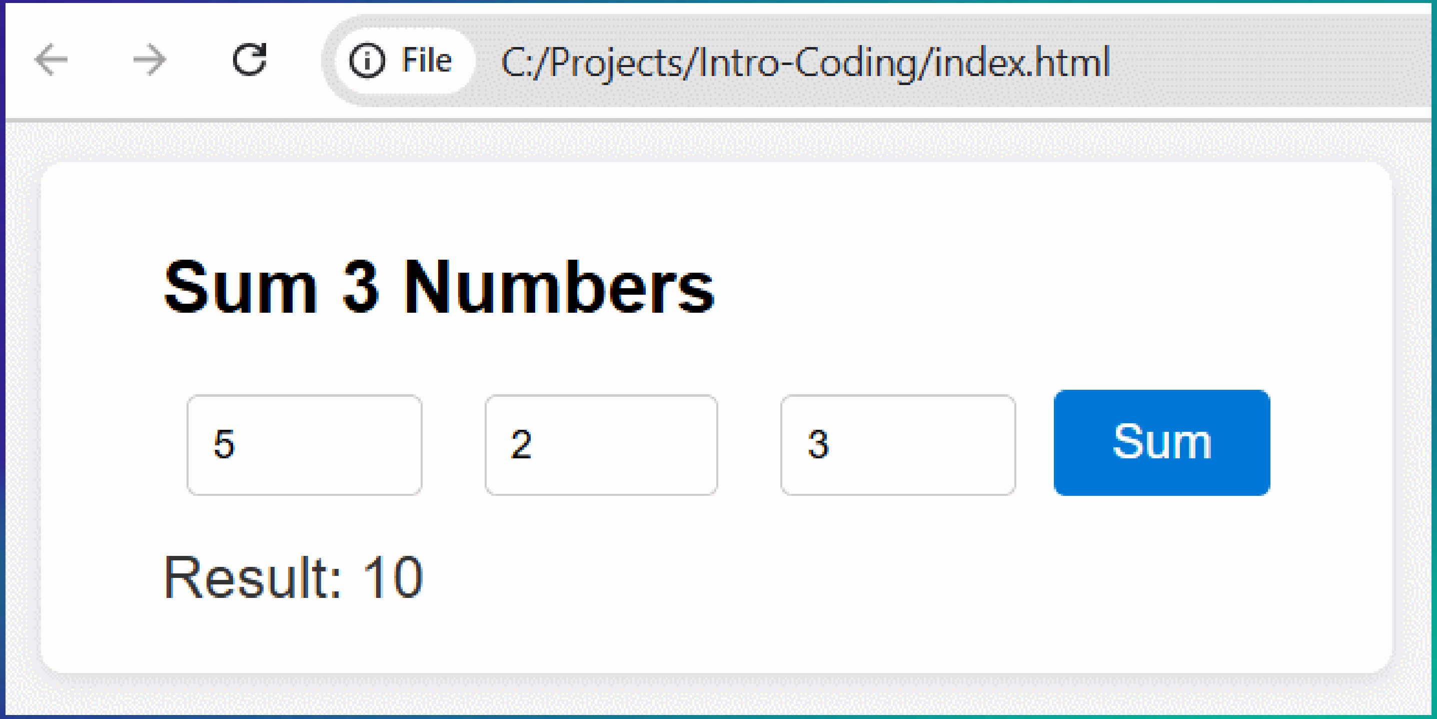
Break

(10 minutes)

[Start timer](#) →

Data, Calculations, Functions

Simple Calculations with JavaScript



Sum 3 Numbers

5 2 3

Result: 10

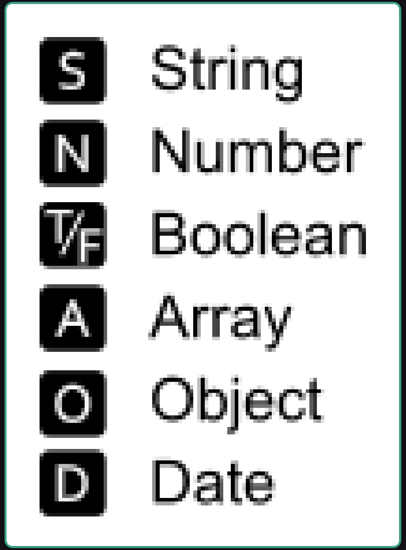
Variables and Data

- **Variables** are **named** locations, which hold **values** (data):

```
let size = 5;
let firstName = "Maria";
```

declaration **name** **value**

- Values have a **type** (data type):
  - Number, e. g. **5**, **3.14**, **-0.03**
  - Text, e. g. **"Hello, JS"**
  - Boolean, e. g. **true** / **false**



S	String
N	Number
T/F	Boolean
A	Array
O	Object
D	Date

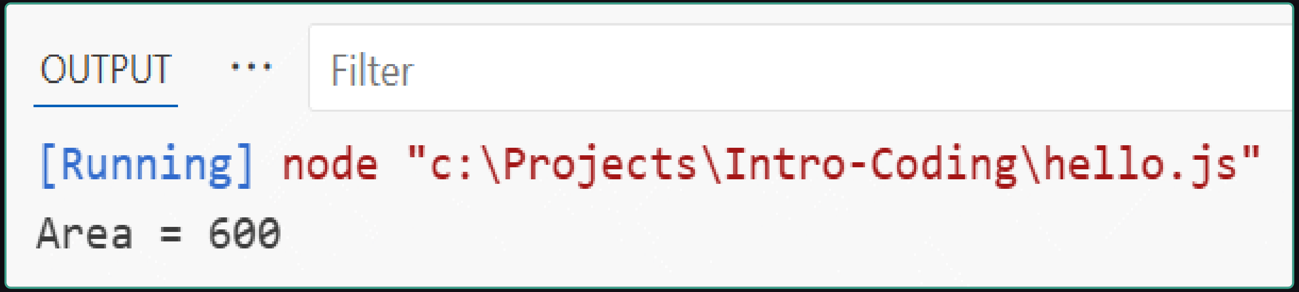
Calculations

- **Calculations** in JavaScript combine variables and values:

```
let width = 20;
let height = 30;
let area = width * height;
```

- **Printing** the result on the console:

```
console.log("Area =", area);
```



```
[Running] node "c:\Projects\Intro-Coding\hello.js"
Area = 600
```

Functions

- **Functions** in JavaScript hold named **pieces of code**:

```
function printRectangle() {
 console.log("----------");
 console.log("|        |");
 console.log("----------");
}
```

```
// Invoke the function
printRectangle();
```

Functions with Parameters

- **Functions** may take **input values** (parameters), and optionally **return a result**

```
function calcArea(width, height) {
  return width * height;
}
```

```
// Invoke the function
let area = calcArea(30, 20);
console.log(area); // 600
```

Problem: Sum 3 Numbers

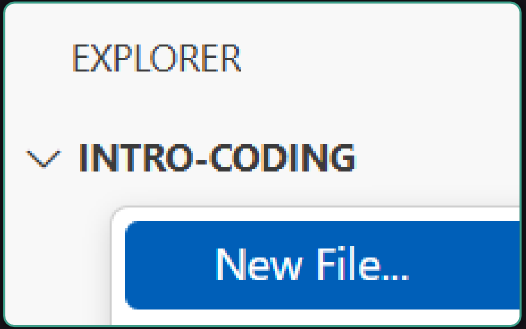
- Write a **function** to **sum 3 numbers**
- Sample usage:

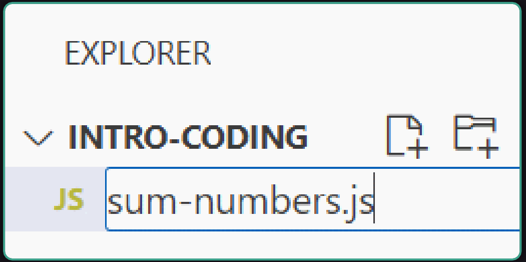
```
let sum = sumNumbers(5, 2, 3);
console.log("Sum =", sum);
// Sum = 10
```

```
let sum = sumNumbers(10, -5, 0.2);
console.log("Sum =", sum);
// Sum = 5.2
```

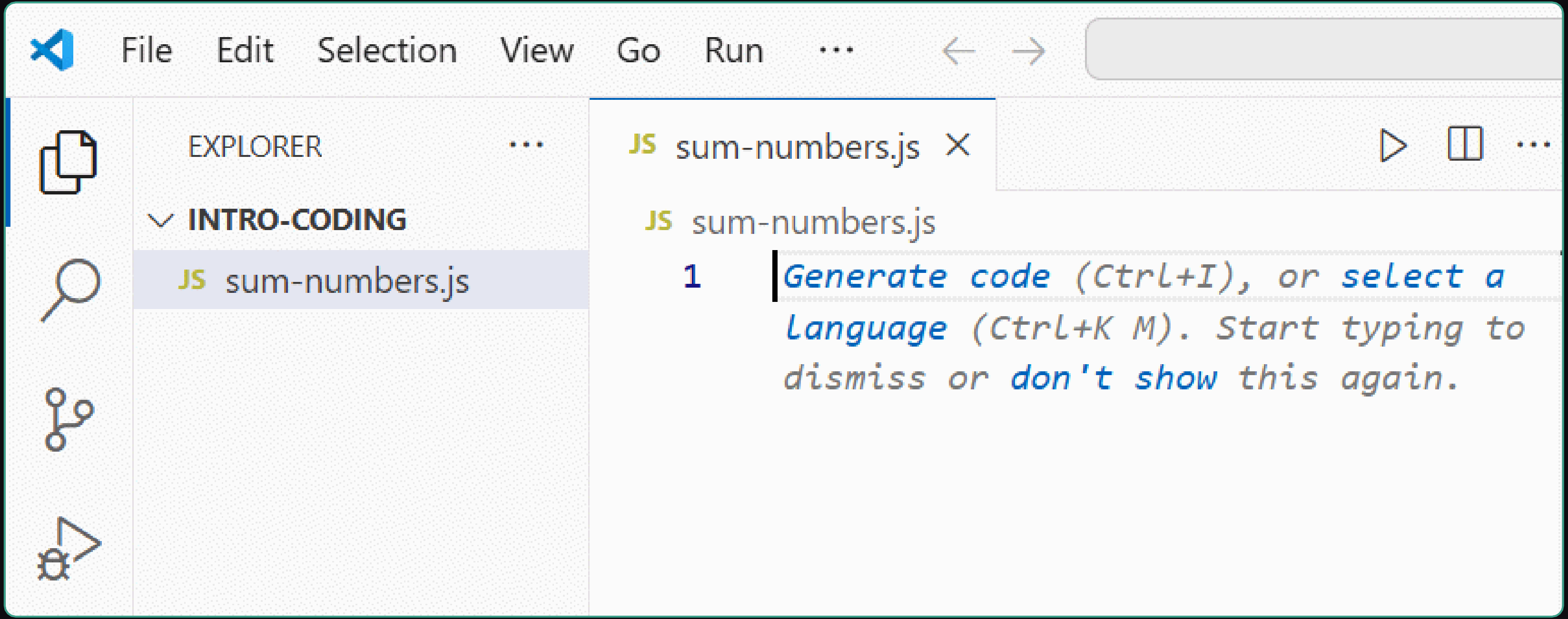
Solution: Sum 3 Numbers

- First, create a **new file** to hold your code
  - Use a **meaningful name** (self-explainable)
  - Example: **sum-numbers.js**





sum-numbers.js



Generate code (Ctrl+I), or select a language (Ctrl+K M). Start typing to dismiss or don't show this again.

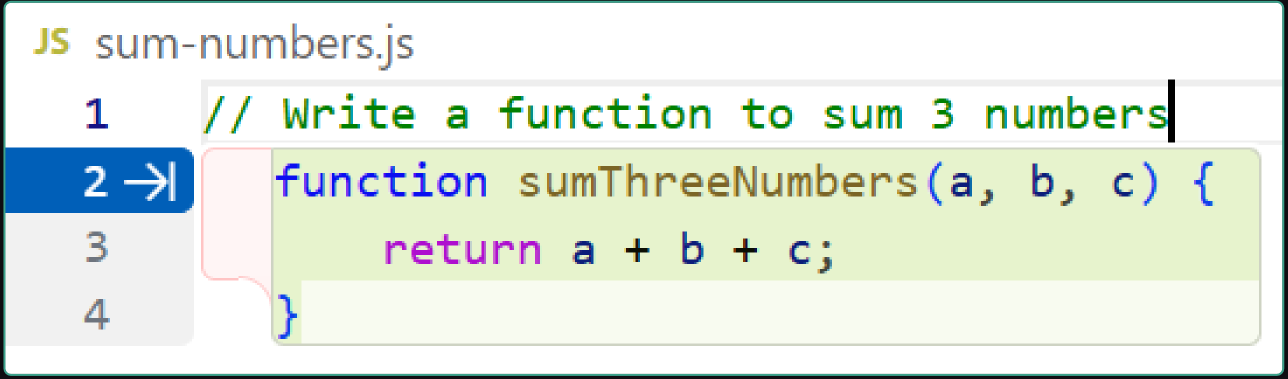
Solution: Sum 3 Numbers (2)

- You can write the function **by hand**:

```
function sumNumbers(a, b, c) {
  return a + b + c;
}
```

- Or ask **Copilot** to write it by typing a **prompt**:

```
// Write a function
to sum 3 numbers
```



sum-numbers.js

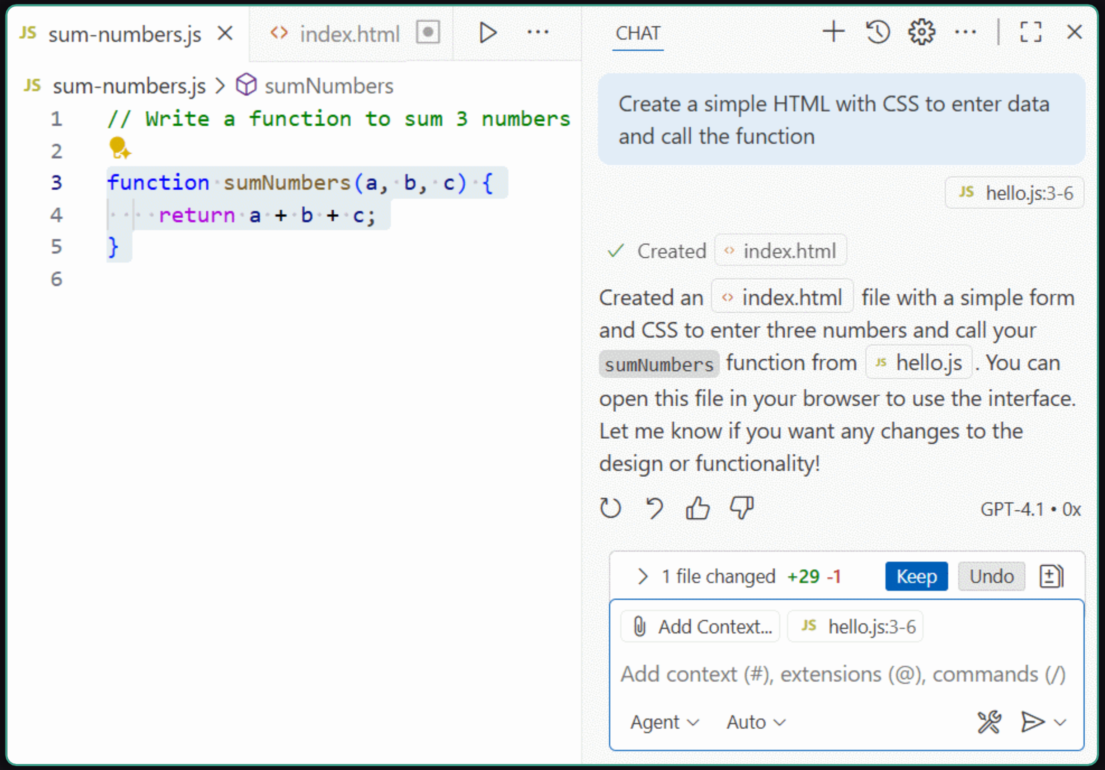
```
// Write a function to sum 3 numbers
  function sumThreeNumbers(a, b, c) {
      return a + b + c;
  }
```

- Press **[Tab]** to accept

Why Do We Need Functions?

- In software **function** implement **actions** or **handle events**
  - E. g. when a button is clicked a function is called
- **Practical use** of functions?
  - This is how to use a function **sumNumbers(a, b, c)** in a simple app (HTML page):

```
Create a simple HTML
with CSS to enter data
and call the function
```



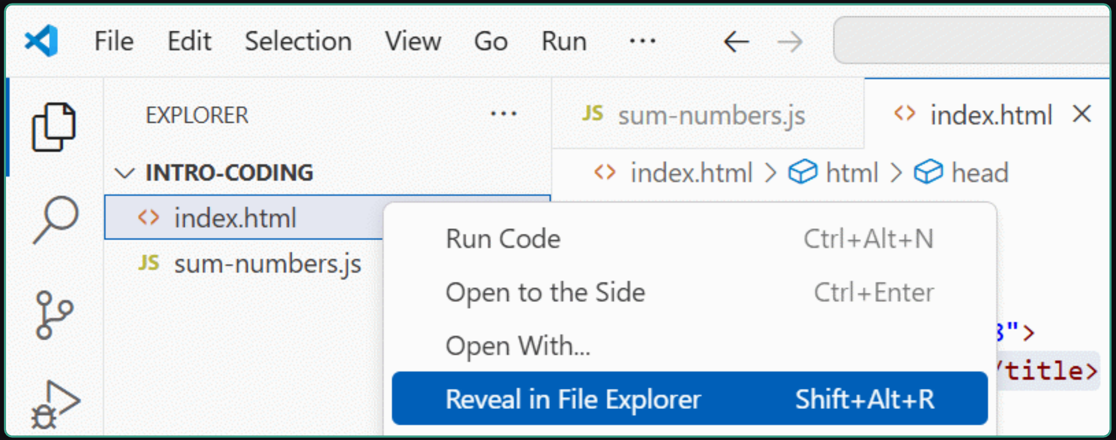
```
// Write a function to sum 3 numbers

function sumNumbers(a, b, c) {
    return a + b + c;
}
```

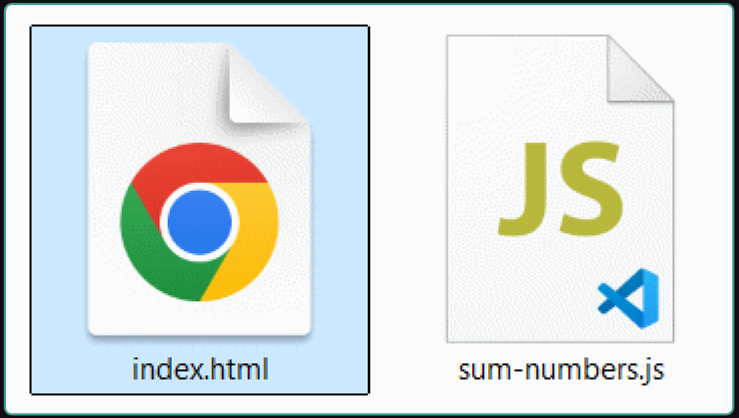
Create a simple HTML with CSS to enter data and call the function

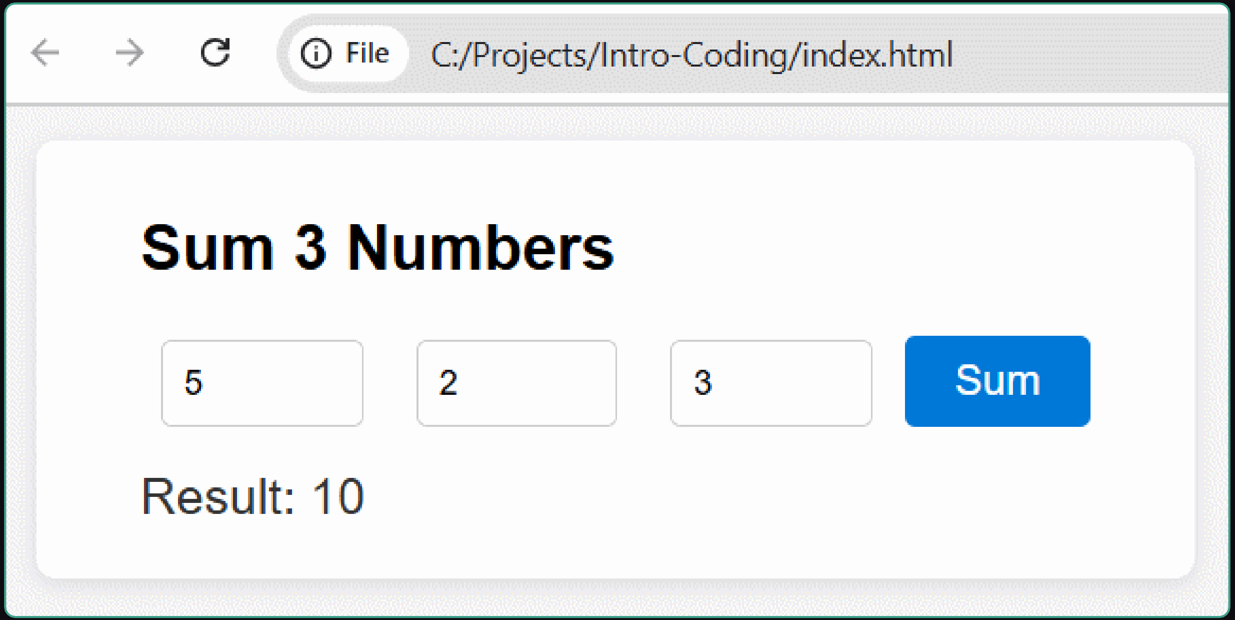
✓ Created index.html

Created an index.html file with a simple form and CSS to enter three numbers and call your `sumNumbers` function from hello.js. You can open this file in your browser to use the interface. Let me know if you want any changes to the design or functionality!

Simple App to Use Your JS Function







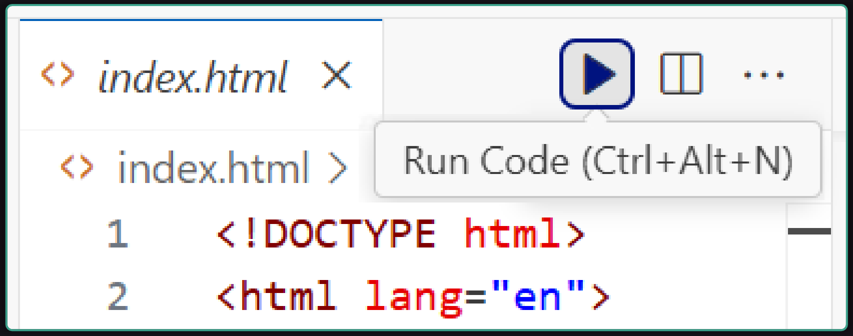
Sum 3 Numbers

5 2 3

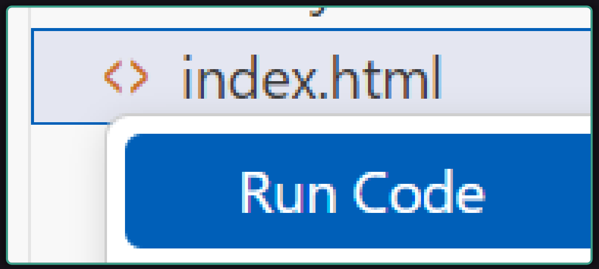
Result: 10

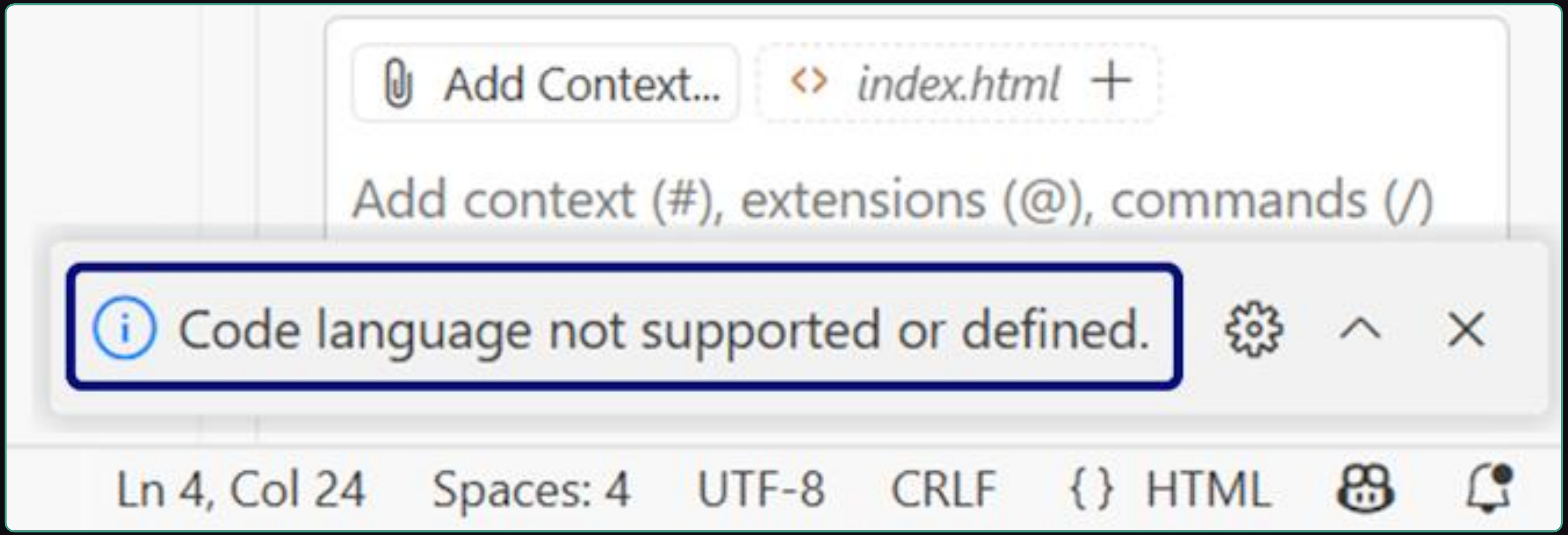
Opening HTML Pages in VS Code

- By default, VS Code and Code Runner extension **cannot run HTML pages**



```
<!DOCTYPE html>
<html lang="en">
```



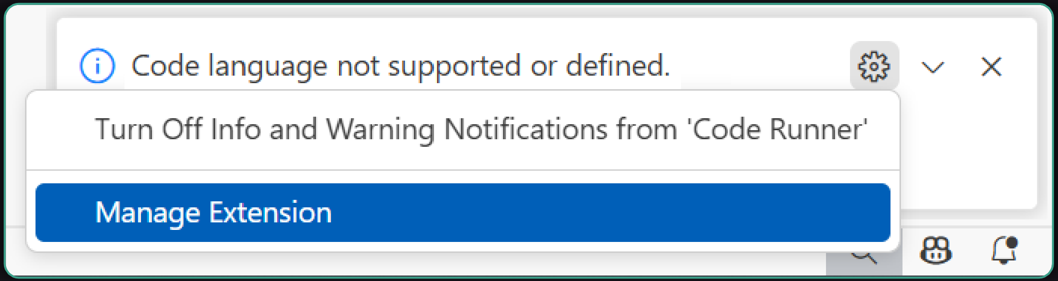


Code language not supported or defined.

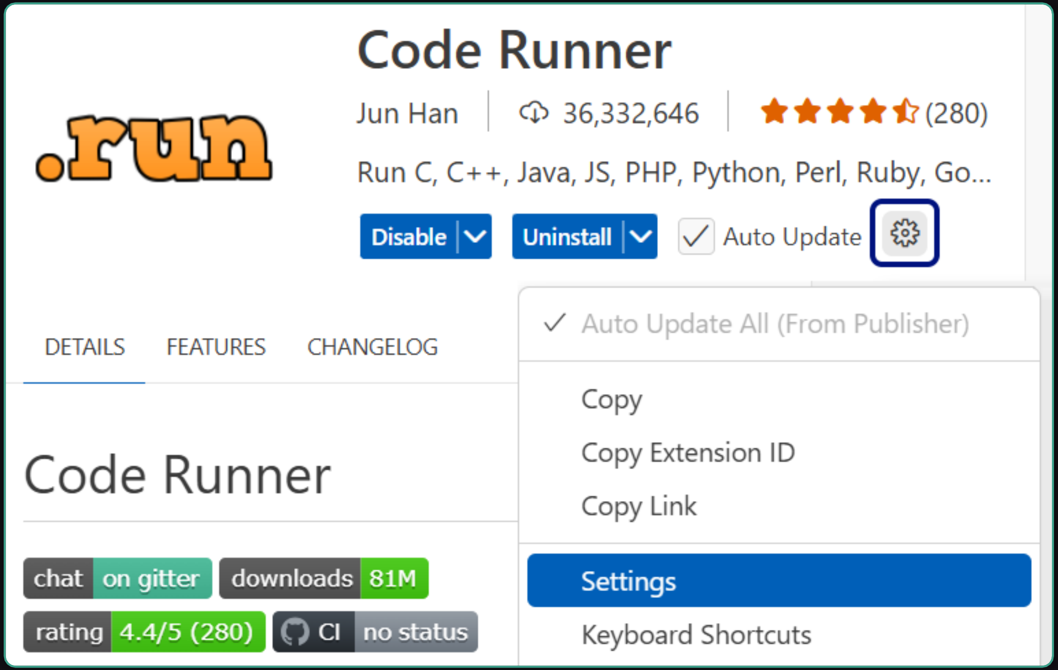
- How to **fix** this?

Configuring Code Runner

- Open Code Runner **extension settings**



Code language not supported or defined.

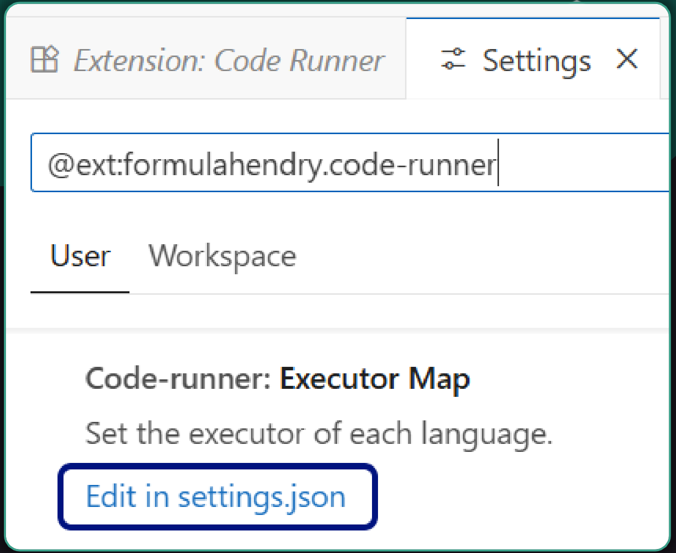


Code Runner

Jun Han | 36,332,646 | ★★★★☆ (280)

Run C, C++, Java, JS, PHP, Python, Perl, Ruby, Go...

- Open the **Executor Map**

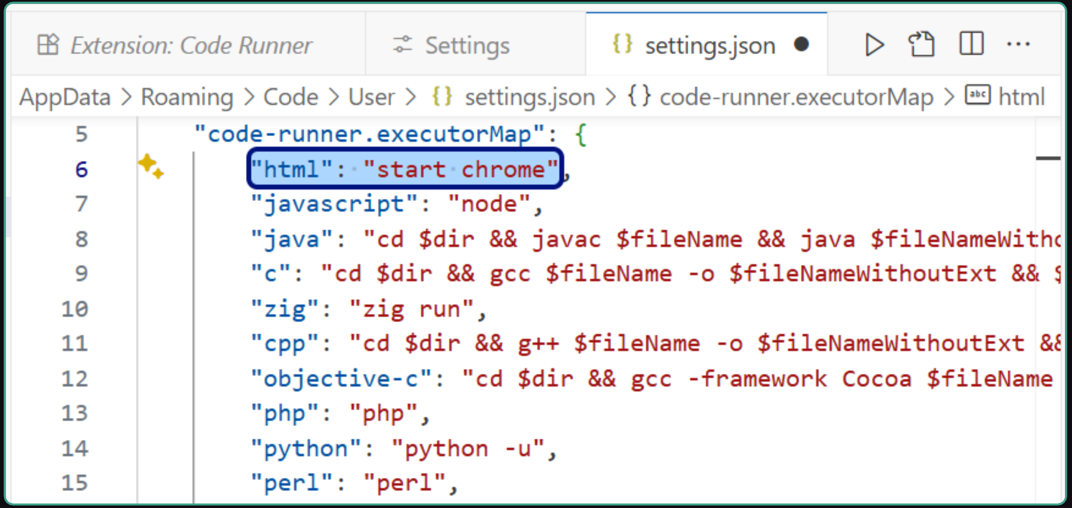


@ext:formulahendry.code-runner

Code-runner: Executor Map

Set the executor of each language.

- Add "**html**" files to open in **Chrome** Web browser:

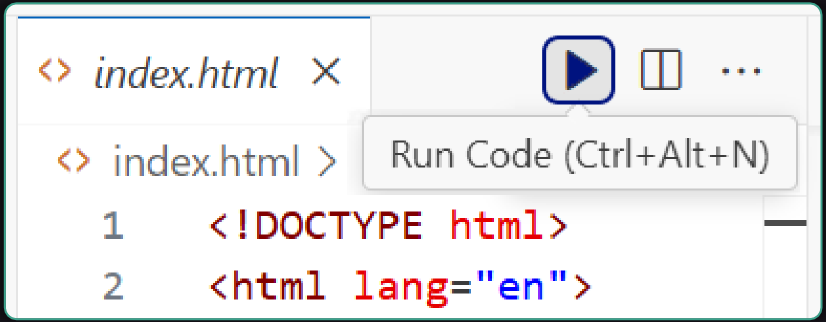


Extension: Code Runner
Settings
settings.json
AppData > Roaming > Code > User > settings.json > code-runner.executorMap > html

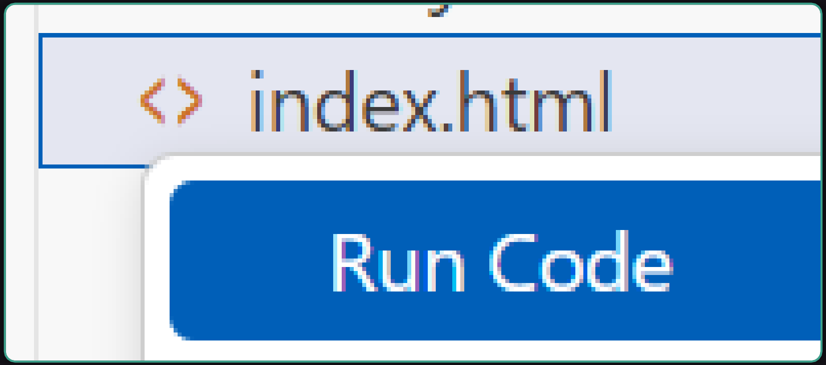
```
    "code-runner.executorMap": {
        "html": "start chrome",
        "javascript": "node",
        "java": "cd $dir && javac $fileName && java $fileNameWitho
        "c": "cd $dir && gcc $fileName -o $fileNameWithoutExt && $
        "zig": "zig run",
        "cpp": "cd $dir && g++ $fileName -o $fileNameWithoutExt &&
        "objective-c": "cd $dir && gcc -framework Cocoa $fileName
        "php": "php",
        "python": "python -u",
        "perl": "perl",
```

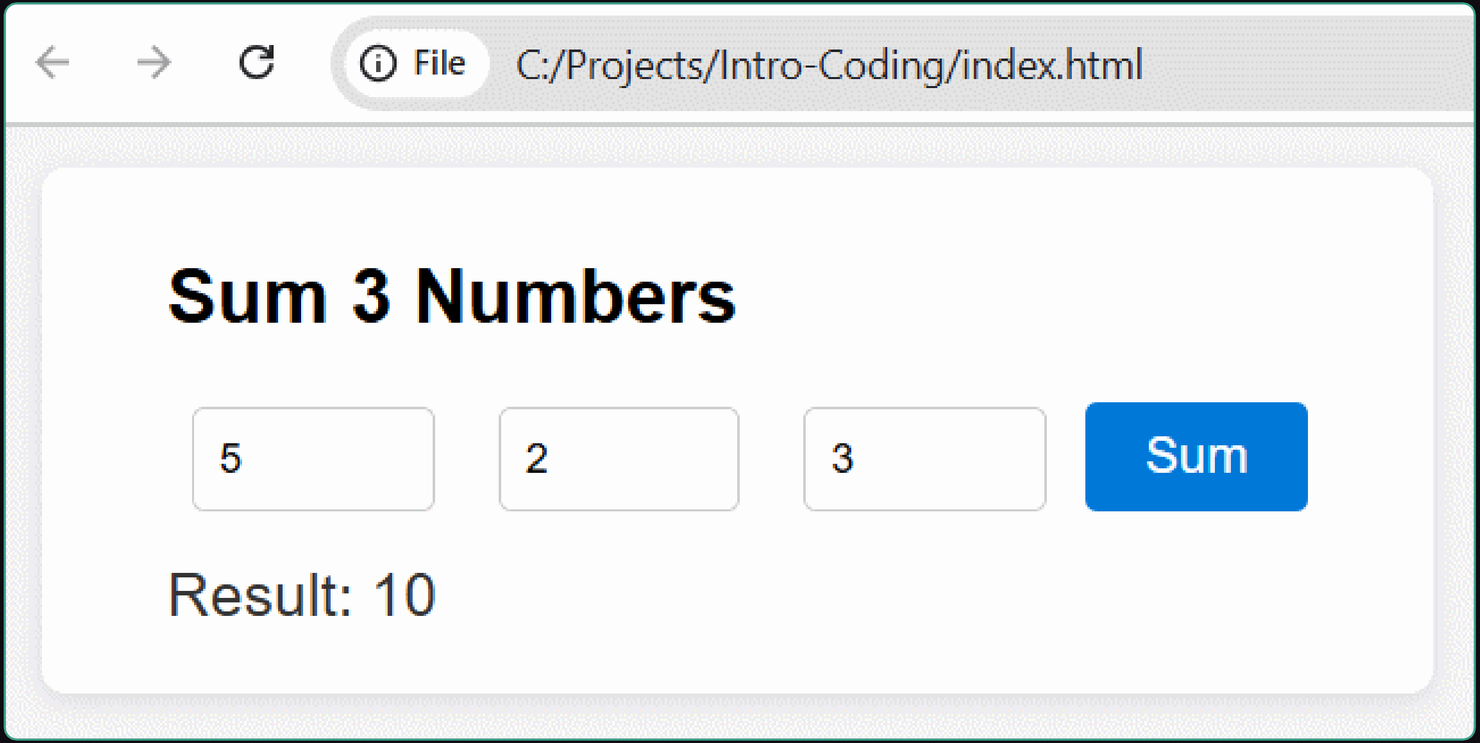
Running HTML Pages

- Now Coder Runner in VS Code will **open Web pages** in the Web browser, when **[Run Code]** is invoked:



```
<!DOCTYPE html>
<html lang="en">
```





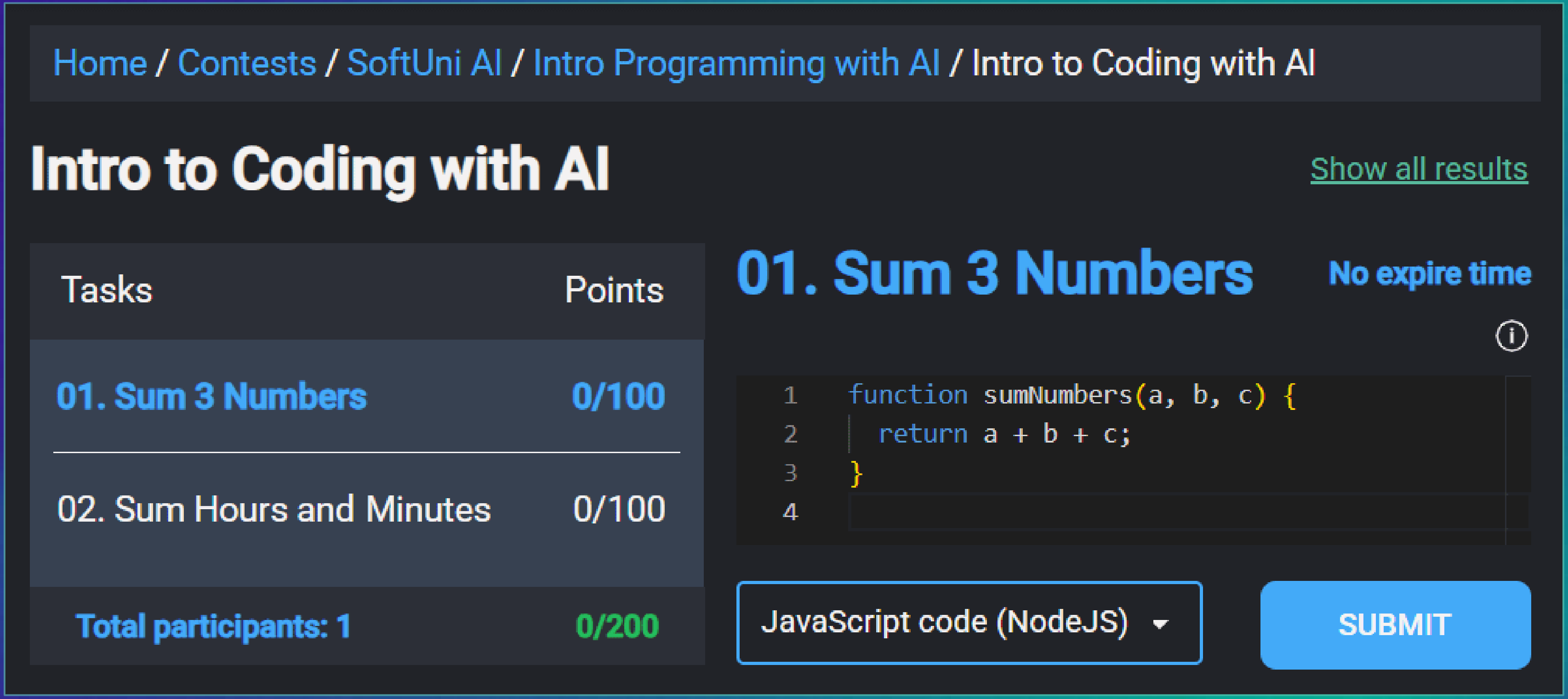
Sum 3 Numbers

5 | 2 | 3 | Sum

Result: 10

SoftUni Judge

Automated Evaluation of Coding Problems



Home / Contests / SoftUni AI / Intro Programming with AI / Intro to Coding with AI

Intro to Coding with AI

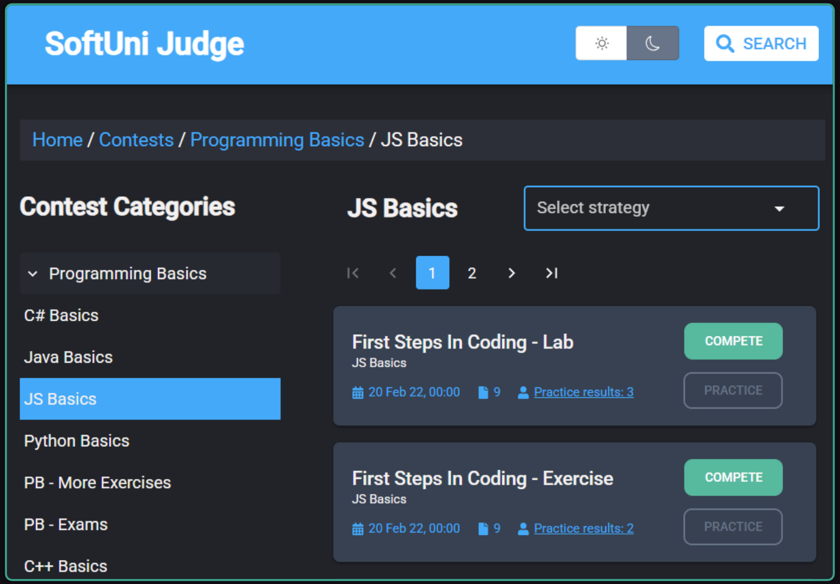
Tasks	Points
01. Sum 3 Numbers	0/100
02. Sum Hours and Minutes	0/100
Total participants: 1	0/200

01. Sum 3 Numbers

```
function sumNumbers(a, b, c) {
  return a + b + c;
}

```

SoftUni Judge

- **SoftUni Judge** is an automated code **evaluation system** (automated grader), used for checking exercises / exams
  - Students submit their code for **evaluation** and receive **real-time result** → correct / incorrect / partially correct
- Each problem is evaluated with several pre-prepared **test cases**
  - **Input** data → expected **output** data
  - When a test **fails**, students can see **what's wrong** (for open tests only)



Home / Contests / Programming Basics / JS Basics

Contest Categories

- Programming Basics
- C# Basics
- Java Basics
- JS Basics
- Python Basics
- PB - More Exercises
- PB - Exams
- C++ Basics

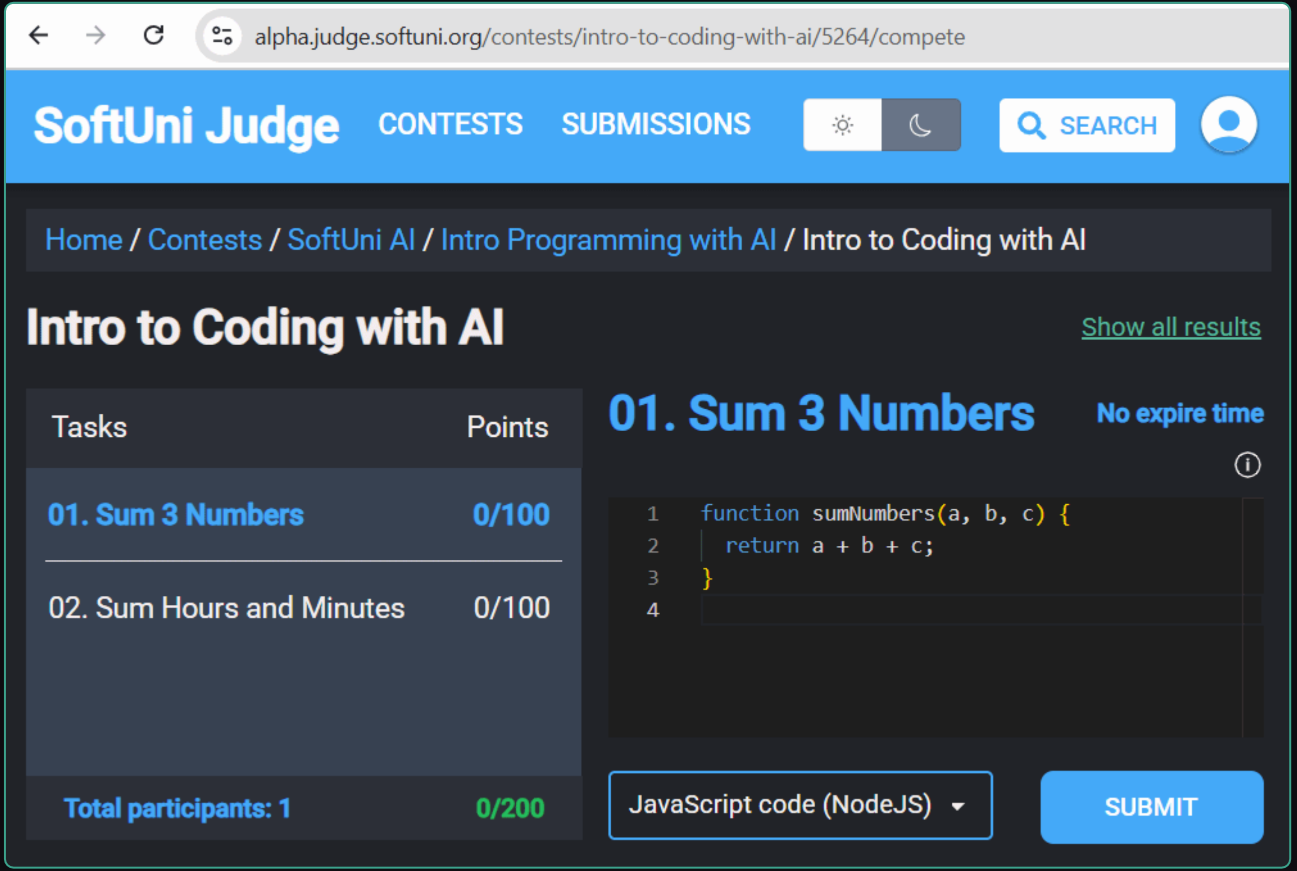
JS Basics

First Steps In Coding - Lab
JS Basics
20 Feb 22, 00:00 9 Practice results: 3

First Steps In Coding - Exercise
JS Basics
20 Feb 22, 00:00 9 Practice results: 2

Submitting a Solution to the Judge

- Submit the function **sumNumbers(a, b, c)** in this contest:
  https://alpha.judge.softuni.org/contests/intro-to-coding-with-ai/5264


alpha.judge.softuni.org/contests/intro-to-coding-with-ai/5264/compete
SoftUni Judge
CONTESTS
SUBMISSIONS
SEARCH

Home / Contests / SoftUni AI / Intro Programming with AI / Intro to Coding with AI

Intro to Coding with AI

Show all results

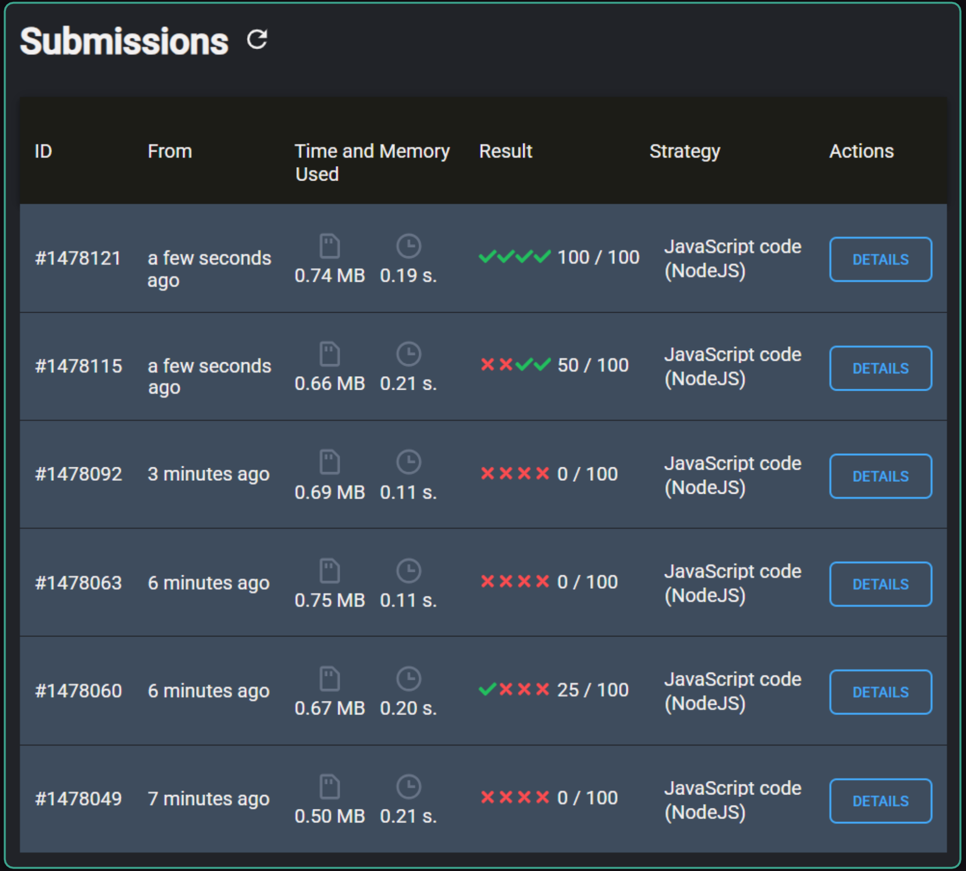
Tasks	Points
01. Sum 3 Numbers	0/100
02. Sum Hours and Minutes	0/100
Total participants: 1	0/200

01. Sum 3 Numbers

No expire time

```
function sumNumbers(a, b, c) {
  return a + b + c;
}

```

JavaScript code (NodeJS)
SUBMIT



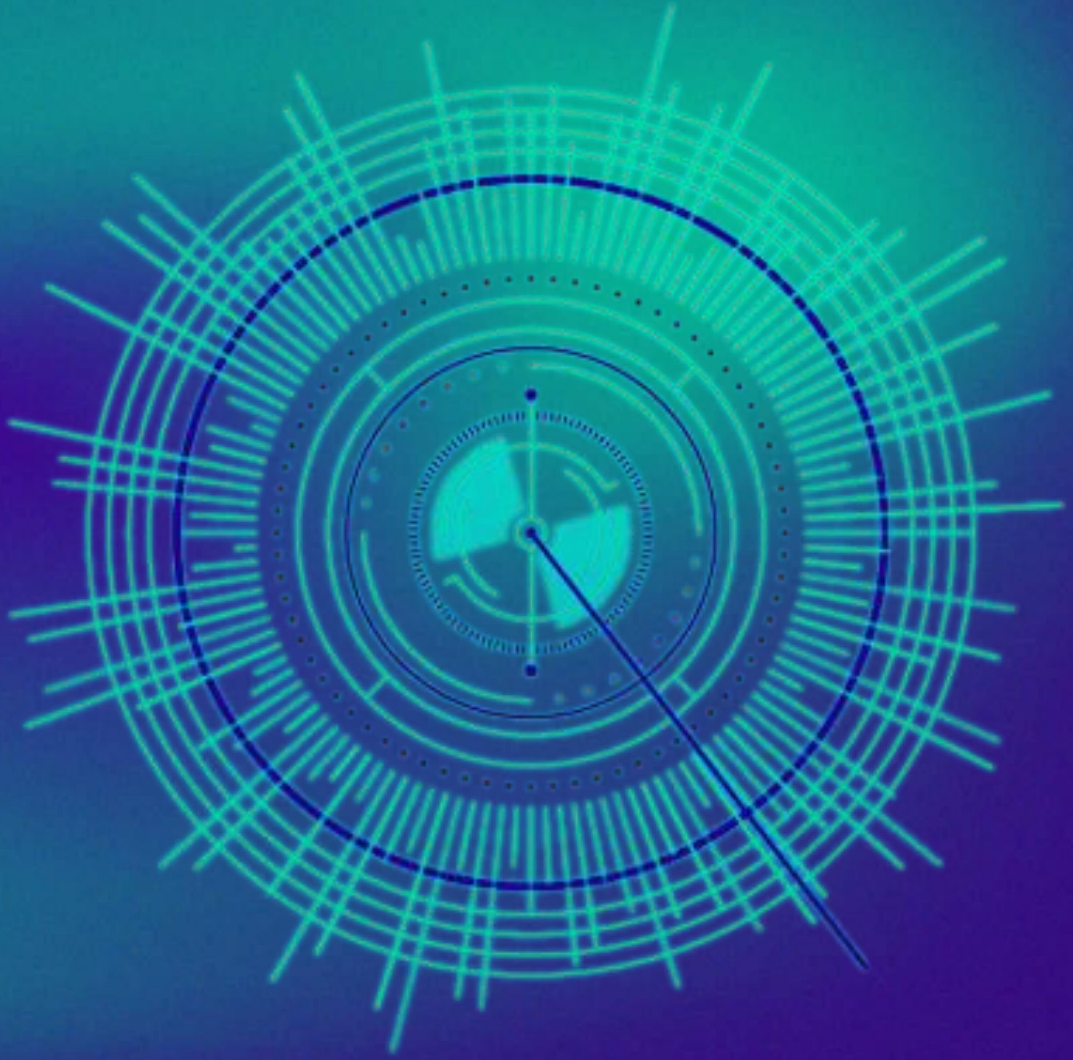
Submissions

ID	From	Time and Memory Used	Result	Strategy	Actions
#1478121	a few seconds ago	0.74 MB 0.19 s.	✓✓✓✓ 100 / 100	JavaScript code (NodeJS)	DETAILS
#1478115	a few seconds ago	0.66 MB 0.21 s.	✗✗✓✓ 50 / 100	JavaScript code (NodeJS)	DETAILS
#1478092	3 minutes ago	0.69 MB 0.11 s.	✗✗✗✗ 0 / 100	JavaScript code (NodeJS)	DETAILS
#1478063	6 minutes ago	0.75 MB 0.11 s.	✗✗✗✗ 0 / 100	JavaScript code (NodeJS)	DETAILS
#1478060	6 minutes ago	0.67 MB 0.20 s.	✓✗✗✗ 25 / 100	JavaScript code (NodeJS)	DETAILS
#1478049	7 minutes ago	0.50 MB 0.21 s.	✗✗✗✗ 0 / 100	JavaScript code (NodeJS)	DETAILS

Break

(10 minutes)

[Start timer](#) →



Simple Operations

Working with Text and Numbers in JS

Concatenate Strings and Numbers

- **Concatenating** (joining) strings and numbers:

```
let firstName = "Steve";
let lastName = "Nak";
let age = 19;

let str = firstName + " " + lastName + " @ " + age;
console.log(str); // Steve Nak @ 19

let text = `${firstName} is ${age} years old.`;
console.log(text); // Steve is 19 years old.
```

Arithmetical Calculations

- **Arithmetical** calculations in JS: **+**, **-**, *****, **/**, **%**, ******, **()**

```
let a = 5, b = 3;
console.log(a + b);  // 8
console.log(a - b);  // 2
console.log(a * b);  // 15
console.log(a / b);  // 1.6666666666666667
console.log(a % b);  // 2
console.log(a ** b); // 125
let expr = a + (b - a) * 0.25;
console.log(expr);  // 4.5
```

Short Calculations

- Shorthand operators:
  - **++**, **--** (prefix, postfix)
  - **+=**, **-=**, ***=**, **/=**, **%=**

```
let sum = 0;
sum += 10;
console.log(sum);      // 10
console.log(sum *= 2); // 20
console.log(sum);      // 20
console.log(sum % 12); // 8
```

```
let count = 0;
count++; // 1
++count; // 2
console.log(count);   // 2
console.log(count++); // 2
console.log(count);   // 3
console.log(++count); // 4
console.log(count);   // 4
console.log(--count); // 3
console.log(count);   // 3
```

Const, Let and Var

- 3 ways to **declare a variable** in JS:
  - **`let`** – declares a block-scoped variable

```
{ // block start
  let x = 5;
  console.log(x); // 5
} // block end
console.log(x);
//  ReferenceError: x is not defined
```

  - **`const`** – a read-only block-scoped variable

```
const c = 5;
console.log(c); // 5
c = c + 5;
// Error: Assignment to
constant variable.
```

  - **`var`** –old-fashioned function-scoped variable (deprecated)

Problem: Sum Hours and Minutes

- Write a **function** to **sum times** in 24-hour format: **hh:mm**
- Examples:

```
sumTimes(
  10, 30, 0, 5);
```

```
The time is 10:30 now.
After 00:05 the time will be 10:35.
```

```
sumTimes(
  10, 30, 0, 45);
```

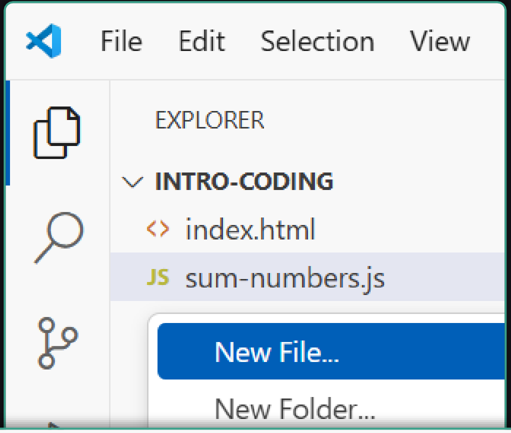
```
The time is 10:30 now.
After 00:45 the time will be 11:15.
```

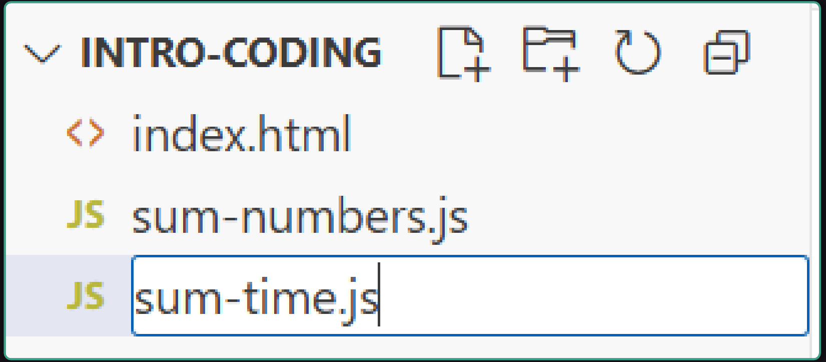
```
sumTimes(
  23, 58, 1, 15);
```

```
The time is 23:58 now.
After 01:15 the time will be 01:13.
```

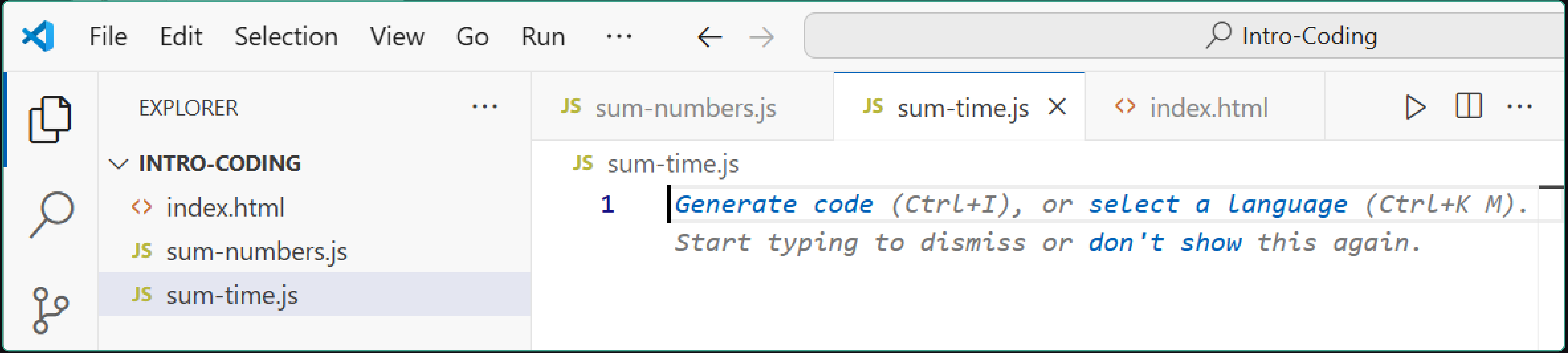
Solution: Sum Hours and Minutes

- First, create a new file "**sum-time.js**"





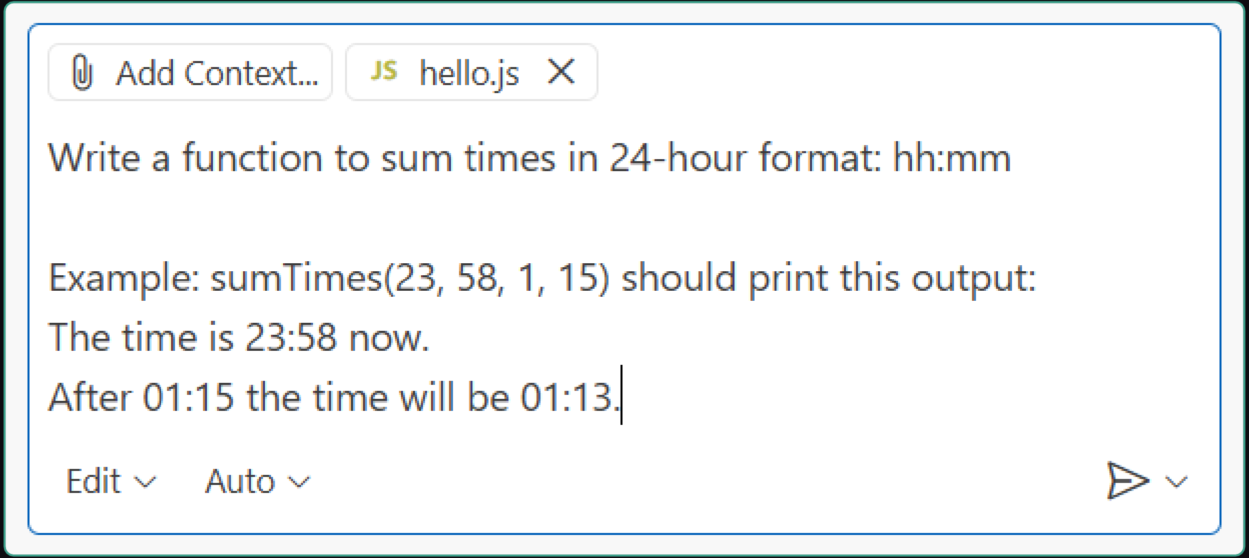
sum-time.js



Generate code (Ctrl+I), or select a language (Ctrl+K M). Start typing to dismiss or don't show this again.

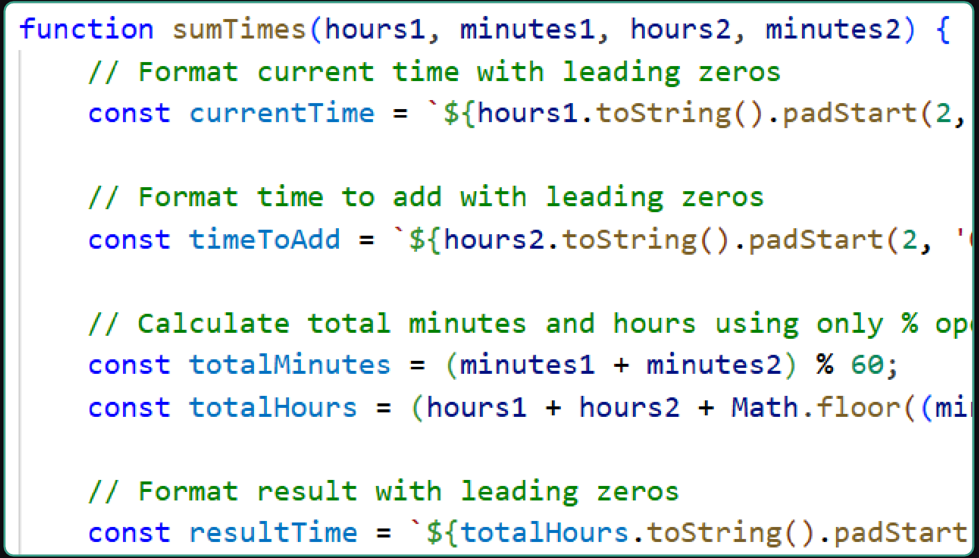
Solution: Sum Hours and Minutes (2)

- Use **GitHub Copilot** in **[Edit] mode** to write the function:

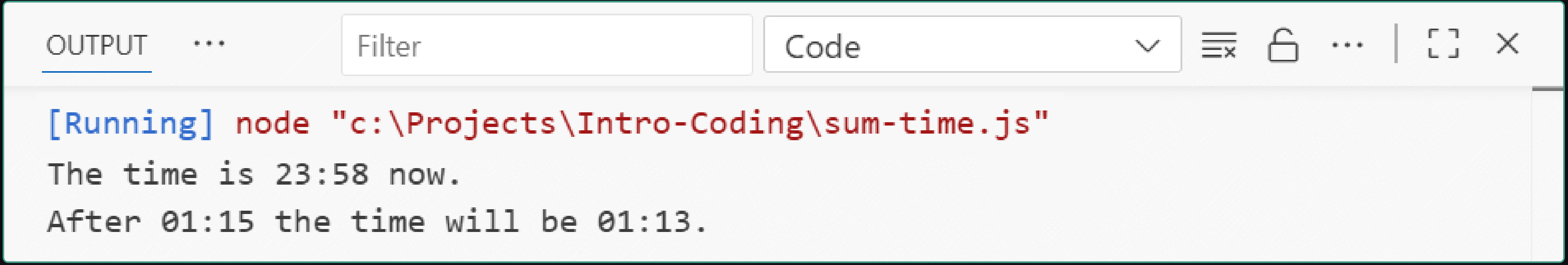


Write a function to sum times in 24-hour format: hh:mm

Example: sumTimes(23, 58, 1, 15) should print this output:
The time is 23:58 now.
After 01:15 the time will be 01:13.



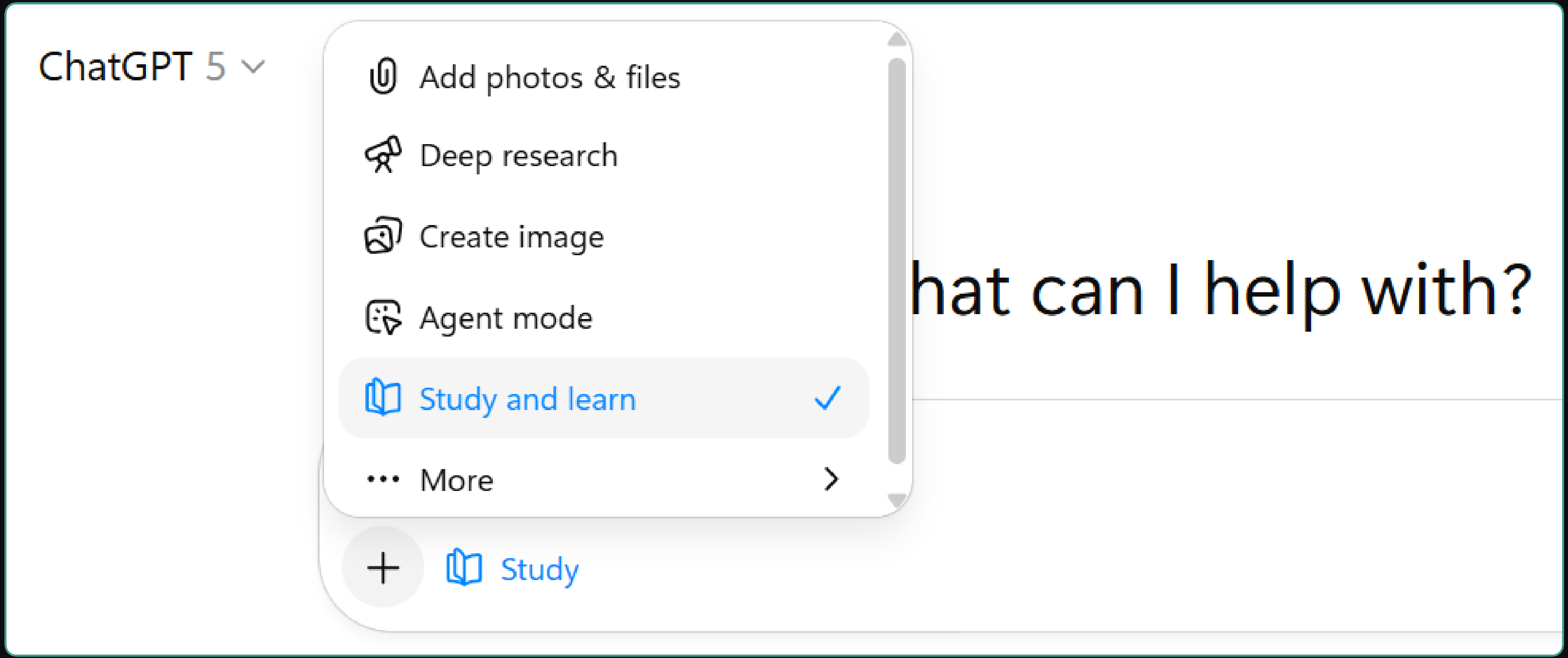
```
function sumTimes(hours1, minutes1, hours2, minutes2) {
    // Format current time with leading zeros
    const currentTime = `${hours1.toString().padStart(2,

    // Format time to add with leading zeros
    const timeToAdd = `${hours2.toString().padStart(2, '

    // Calculate total minutes and hours using only % op
    const totalMinutes = (minutes1 + minutes2) % 60;
    const totalHours = (hours1 + hours2 + Math.floor((mi

    // Format result with leading zeros
    const resultTime = `${totalHours.toString().padStart
```



```
[Running] node "c:\Projects\Intro-Coding\sum-time.js"
The time is 23:58 now.
After 01:15 the time will be 01:13.
```

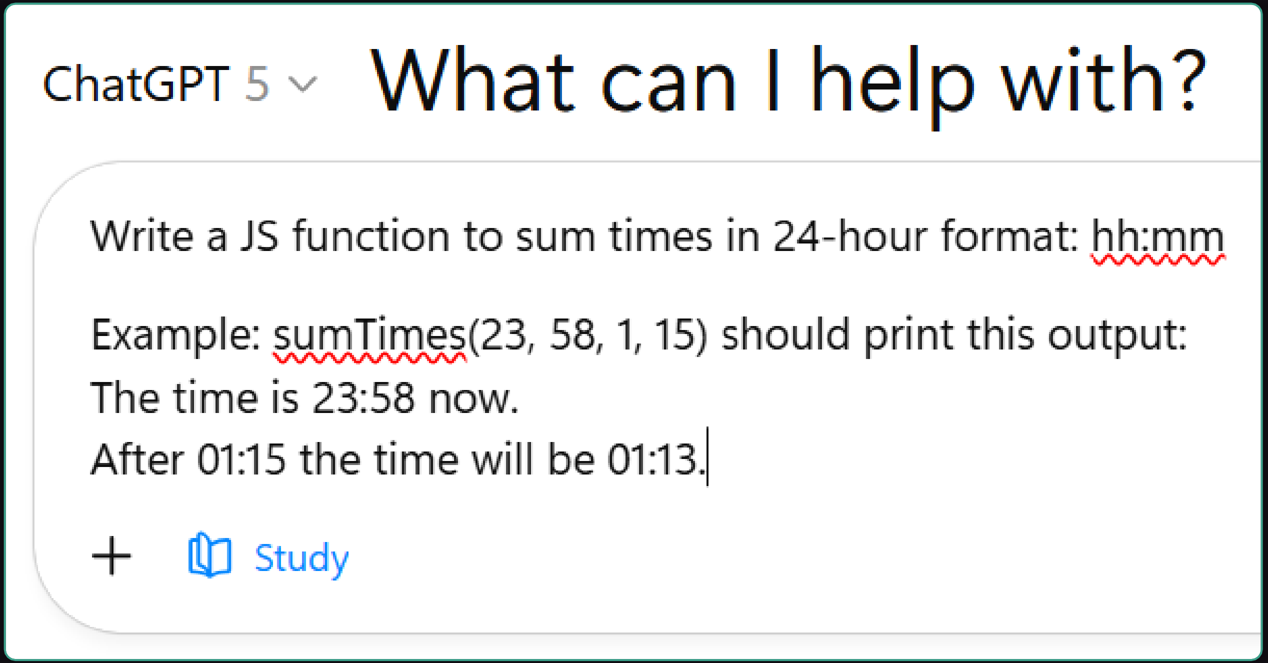
Solution: Sum Hours and Minutes (3)

- Use **ChatGPT Learn Mode** to help you writing the function:



- **"*Study and learn*" mode** in ChatGPT does not directly solve the problem, but **guides you** to the solution (like a **mentor**)

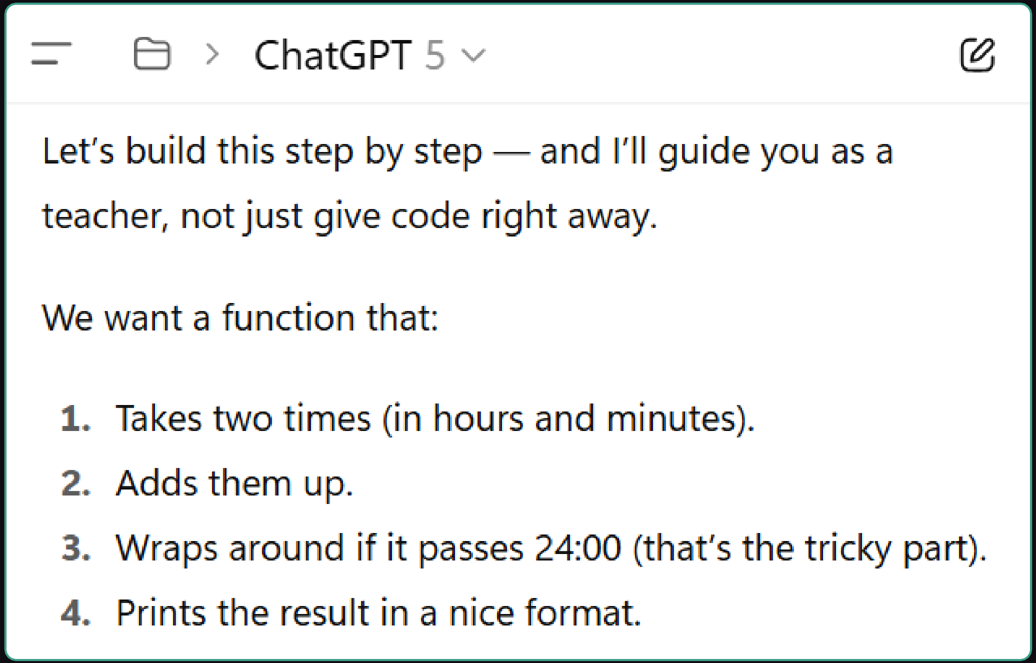
Solution: Sum Hours and Minutes (4)



What can I help with?

Write a JS function to sum times in 24-hour format: hh:mm

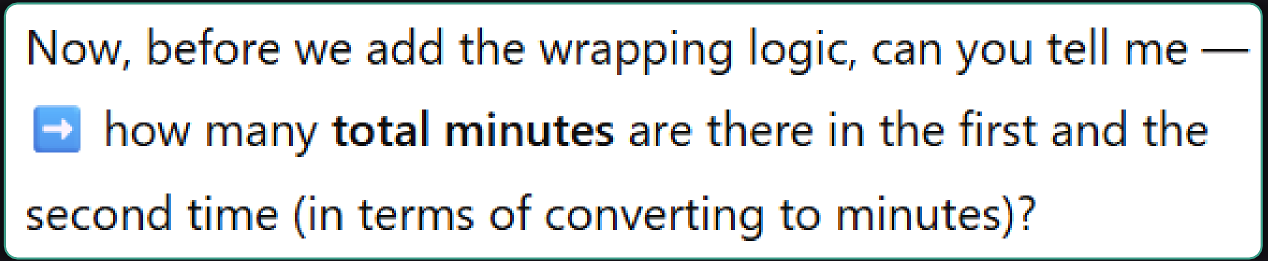
Example: sumTimes(23, 58, 1, 15) should print this output:
The time is 23:58 now.
After 01:15 the time will be 01:13.



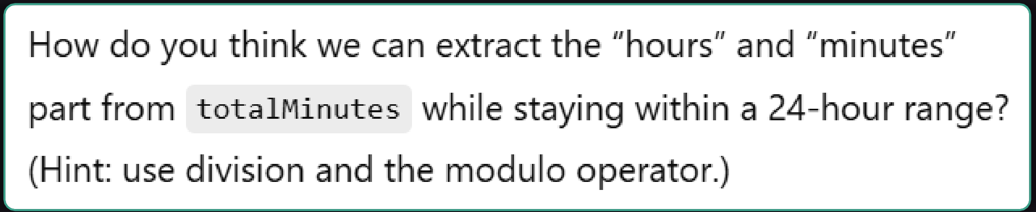
Let's build this step by step — and I'll guide you as a teacher, not just give code right away.

We want a function that:

1. Takes two times (in hours and minutes).
2. Adds them up.
3. Wraps around if it passes 24:00 (that's the tricky part).
4. Prints the result in a nice format.



Now, before we add the wrapping logic, can you tell me —
➡️ how many **total minutes** are there in the first and the second time (in terms of converting to minutes)?

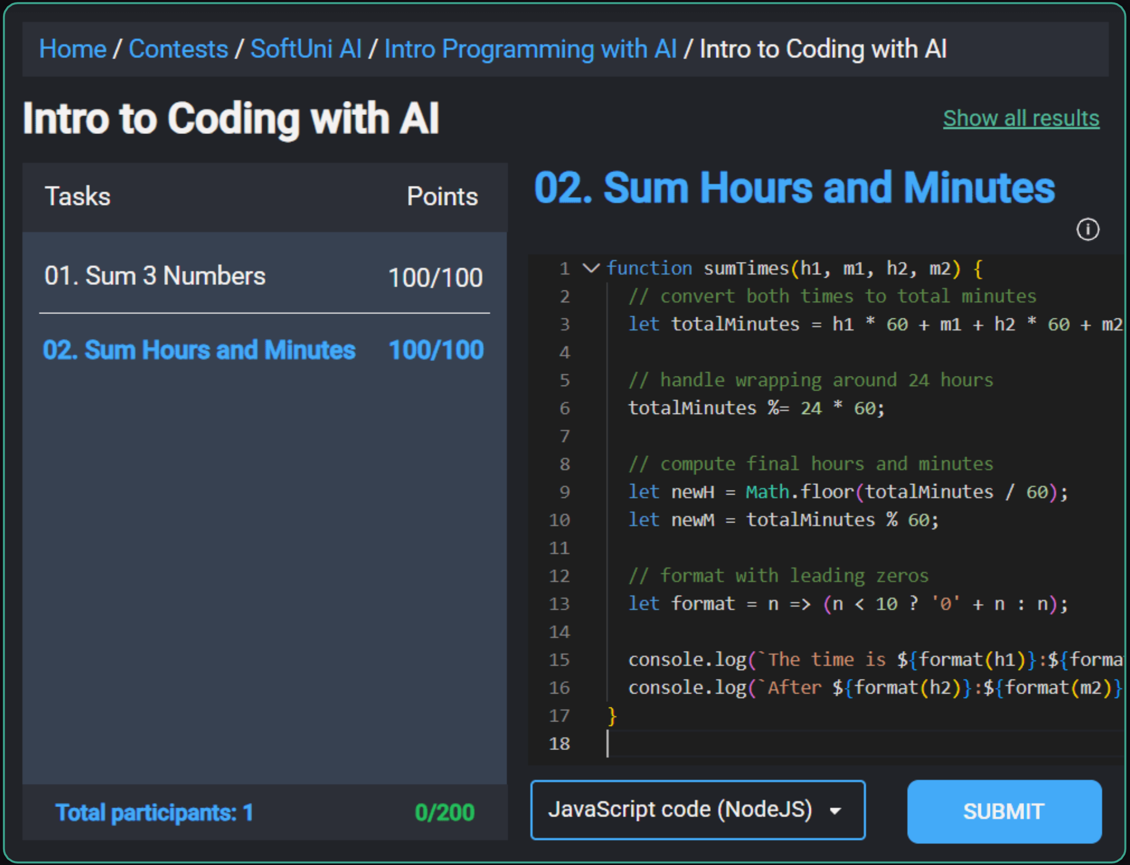


How do you think we can extract the "hours" and "minutes" part from `totalMinutes` while staying within a 24-hour range? (Hint: use division and the modulo operator.)

https://chatgpt.com/share/68e245e4-c770-8013-ae70-dc332d984452

Submitting a Solution to the Judge

- Submit the function **sumTime(h1, m1, h2, m2)** in this contest: https://alpha.judge.softuni.org/contests/intro-to-coding-with-ai/5264



Home / Contests / SoftUni AI / Intro Programming with AI / Intro to Coding with AI

Intro to Coding with AI

Show all results

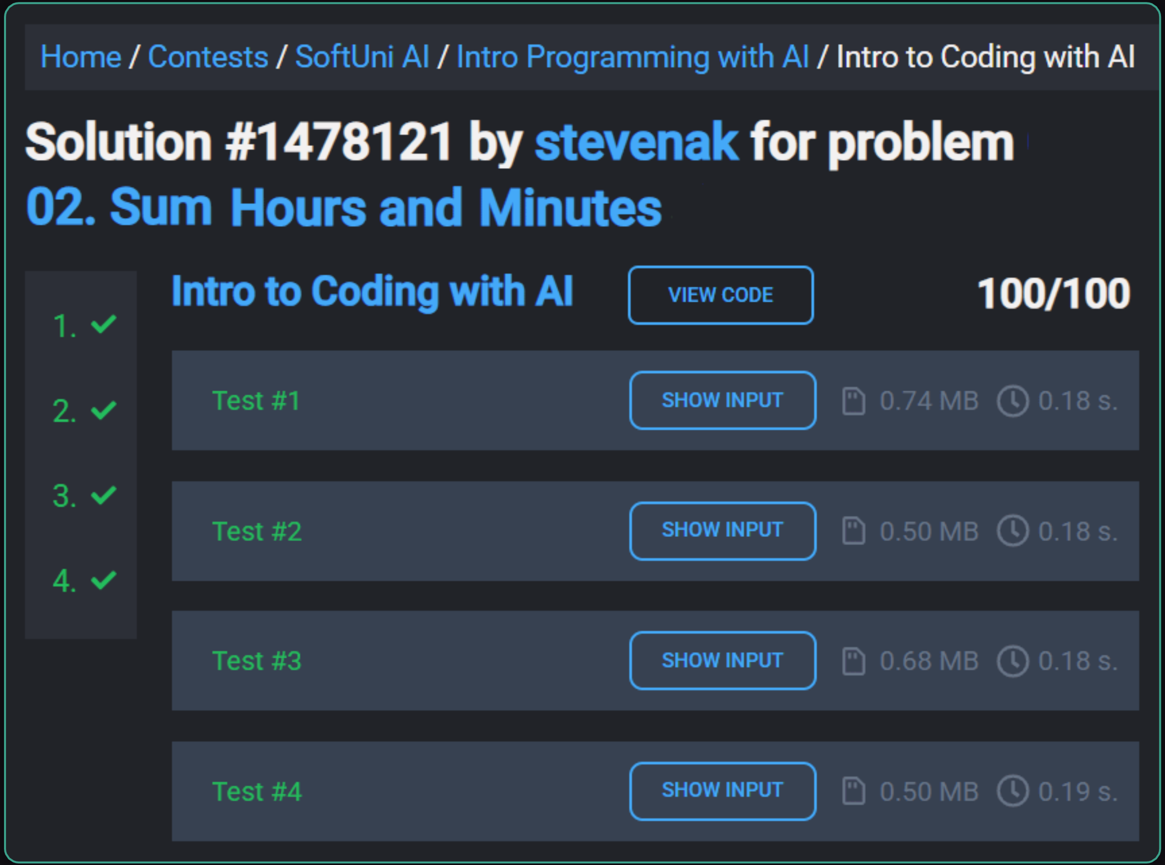
Tasks	Points
01. Sum 3 Numbers	100/100
02. Sum Hours and Minutes	100/100

Total participants: 1 0/200

02. Sum Hours and Minutes

```
function sumTimes(h1, m1, h2, m2) {
  // convert both times to total minutes
  let totalMinutes = h1 * 60 + m1 + h2 * 60 + m2

  // handle wrapping around 24 hours
  totalMinutes %= 24 * 60;

  // compute final hours and minutes
  let newH = Math.floor(totalMinutes / 60);
  let newM = totalMinutes % 60;

  // format with leading zeros
  let format = n => (n < 10 ? '0' + n : n);

  console.log(`The time is ${format(h1)}:${forma
  console.log(`After ${format(h2)}:${format(m2)}
}
```

JavaScript code (NodeJS)



Home / Contests / SoftUni AI / Intro Programming with AI / Intro to Coding with AI

Solution #1478121 by stevenak for problem 02. Sum Hours and Minutes

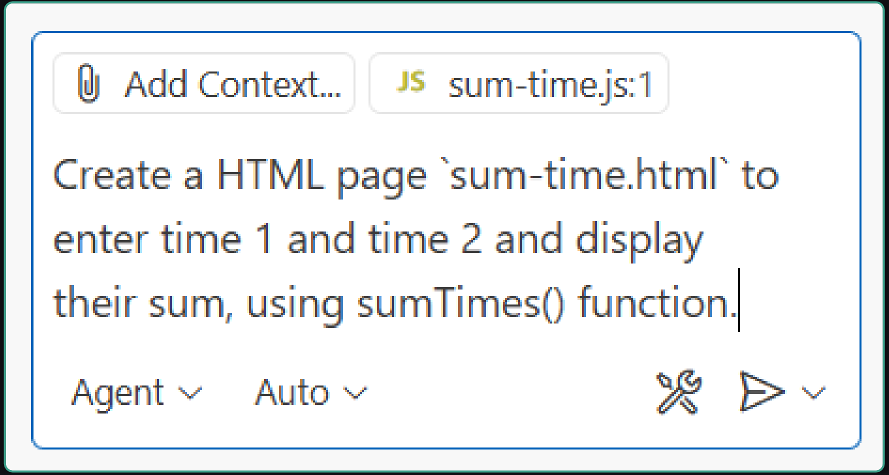
Intro to Coding with AI 100/100

1. ✓	Test #1	0.74 MB	0.18 s.
2. ✓	Test #2	0.50 MB	0.18 s.
3. ✓	Test #3	0.68 MB	0.18 s.
4. ✓	Test #4	0.50 MB	0.19 s.

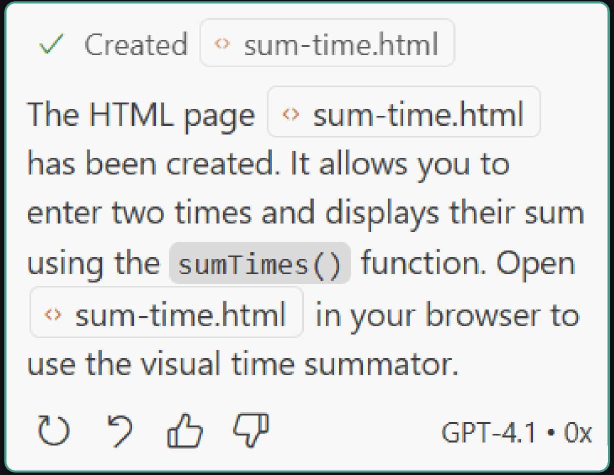
Visualize the Time Summator

- Create an **HMTL page** to visualize the time summator function: enter time 1 and time 2 and calculate their sum

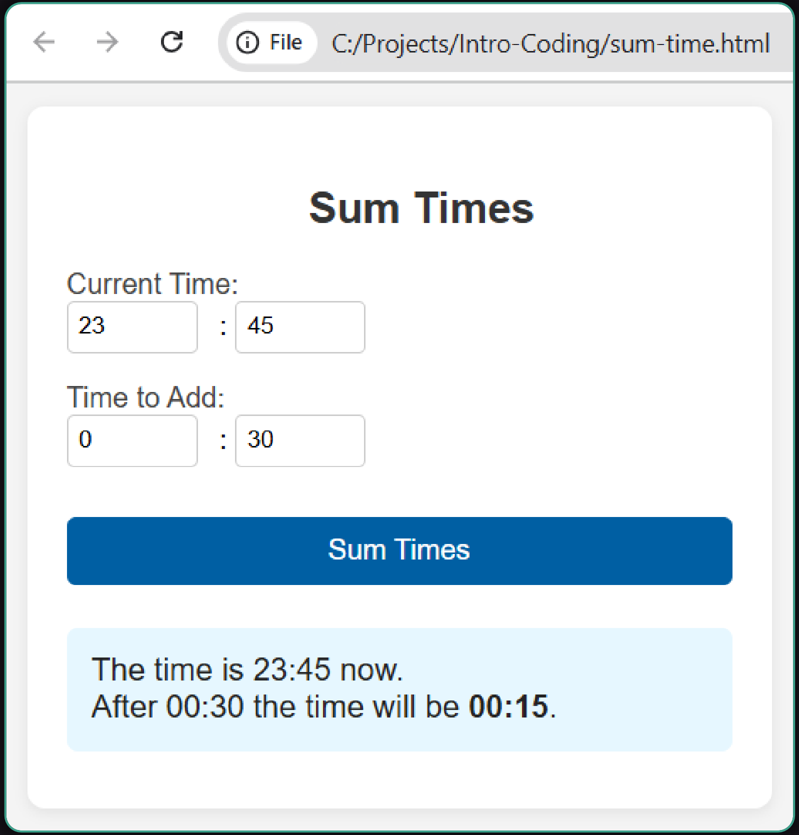
```
Create a HTML page `sum-time.html` to
enter time 1 and time 2 and display
their sum, using sumTimes() function.
```



Create a HTML page `sum-time.html` to enter time 1 and time 2 and display their sum, using sumTimes() function.



The HTML page sum-time.html has been created. It allows you to enter two times and displays their sum using the `sumTimes()` function. Open sum-time.html in your browser to use the visual time summator.



Current Time: 23 : 45

Time to Add: 0 : 30

The time is 23:45 now.
After 00:30 the time will be **00:15**.

Lesson Summary

SoftUni AI

- **Coding** == creating **sequences of commands** in a **programming language** (like JS)
  - **Variables** hold data, used in **expressions** and calculations
  - **Functions** hold named pieces of code, take **parameters**, return a **value**
- **GitHub Copilot** writes functions by text prompt, creates HTML pages and explains how code works
- **SoftUni Judge** evaluates your code → correct / wrong

Questions?

© SoftUni AI – https://ai.softuni.bg. Copyrighted document. Unauthorized copy, reproduction or use is not permitted.

Postbank – Exclusive Partner for SoftUni AI

- One of the leading **banking institutions** in Bulgaria
- Member of the Eurobank Group with € 99.6 billion of assets
- Innovative trendsetter in the digitalization and transformation
- Certified Top Employer 2024 by the international Top Employers Institute
- AI integration for business, learning and development
- AI Assistants for talents: Story Builder, CV Assistant, Interview Trainer
- Proven people care and wellbeing initiatives
- Benefits and unlimited access to professional, personal and leadership trainings and programs
- www.postbank.bg / careers.postbank.bg

Diamond Partners of Software University

Coca-Cola HBC Bulgaria

Postbank

top EMPLOYER
България Bulgaria 2025
FOR A BETTER WORLD OF WORK

SUPER HOSTING .BG

encorp.ai

createX

DRAFT KINGS
THE CROWN IS YOURS

INDEAVR
Serving the high achievers

VIVACOM

Diamond Partners of SoftUni Digital

SoftUni
Digital

Coca-Cola HBC
Bulgaria

Postbank

top EMPLOYER
България
Bulgaria
2025
FOR A BETTER WORLD OF WORK

SUPER
HOSTING
.BG

encorp.ai

DRAFT
KINGS
THE CROWN IS YOURS

createX

INDEAVR
Serving the high achievers

VIVACOM

Diamond Partners of SoftUni Digital

HUMAN

IMPULSE MEDIA®

M

1 FOR FIT

NETPEAK
DIGITAL GROWTH PARTNER

ABC DESIGN &
COMMUNICATION

Zahara
dig.it.all

ETIENYANEV
Break Your *Limits*. Live Your *Brand*.

Aleikov
studio

Diamond Partners of SoftUni Creative

Coca-Cola HBC Bulgaria

Postbank

top EMPLOYER
България
Bulgaria
2025
FOR A BETTER WORLD OF WORK

SUPER HOSTING .BG

encorp.ai

DRAFT KINGS
THE CROWN IS YOURS

createX

INDEAVR
Serving the high achievers

VIVACOM

Organization Partners of SoftUni Creative

dequitas.

Фигмайстор

Studio π

Think Big • Design Smart

дизайнът на нещата

REDV PRINT STUDIO ®

THE BUCKS TOWN'S WORK